X-324-73-80
PREPRINT

NASA TM X- 66209

ATMOSPHERE EXPLORER CONTROL SYSTEM SOFTWARE (VERSION 2.0)

(NASA-TM-X-66209) ATMOSPHERE EXPLORER CONTROL SYSTEM SOFTWARE (VERSION 2.0) (NASA) 93 p HC $6.75 CSCL 22B
N73-20884
Unclas
G3/31 66543

WILLIAM MOCARSKY
ANTHONY VILLASENOR

MARCH 1973

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-324-73-80

ATMOSPHERE EXPLORER CONTROL SYSTEM SOFTWARE

(VERSION 2.0)

William Mocarsky
Anthony Villasenor

March 1973

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

INTRODUCTION

This document describes the basic design of the Atmosphere Explorer Control System (AECS) software used in the testing and integration of the AE spacecraft and experiments. The same basic system is employed by the Test and Evaluation Division at GSFC and the Spacecraft and Test System at RCA, Highstown, New Jersey.

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

ILLUSTRATIONS

ILLUSTRATIONS (continued)

ATMOSPHERE EXPLORER CONTROL SYSTEM SOFTWARE

I. SUMMARY

The Atmosphere Explorer Control System — AECS — has been developed to provide automatic computer control of the AE spacecraft and experiments. The software performs several vital functions, such as issuing commands to the spacecraft and experiments, receiving and processing telemetry data, allowing for extensive data processing by experiment analysis programs, etc.

The AECS has been written for a 48K XEROX Data System Sigma 5 computer, and coexists in core with the XDS Real-time Batch Monitor (RBM) executive system. RBM is a flexible operating system designed for a real-time foreground/background environment, and hence is ideally suited for this application. Existing capabilities of RBM have been used as much as possible by AECS to minimize programming redundancy.

The most important functions of the AECS are to send commands to the spacecraft and experiments, and to receive, process, and display telemetry data. In order to avoid interference between these functions and at the same time permit the execution of other processing and system overhead tasks, the AECS is logically divided into several major processing sections. The duties of each of these sections are as follows:

1. Executive Control Section

The Executive Control Section (ECS) schedules all real-time processing in the computer. Every 100 milliseconds the ECS scans through the system to initiate real-time operations such as the recording of telemetry data on a "history tape," or the updating of visual display screens. All telemetry parameters displayed on the screens can be regularly updated to show the most recently received values. ECS also interprets messages entered through the console keyboards and performs the requested tasks. Such messages can be commands to the spacecraft or experiments, calls for the execution of one or more programs, requests for hardcopy printouts of the display screens, requests for snapshots of selected telemetry data, updates on strip-chart recorders, etc. The ECS also provides the priority scheduling needed to interface real-time input/output operations with the RBM I/O routines.

2. Telemetry Decommutation Section

The Telemetry Decommutation Section (TMD) receives and processes telemetry data at the rate of 16,384 bits per second. The term "main frame" designates the 128 8-bit telemetry words that repeat each 1/16 of a second, beginning with the frame sync pattern (1111 1010 1111 0011 0010 0000). The words are designated 1 through 128. The term main frame applies to these words independent of the subcom counter, which ranges in value from 0 through 127 and defines the 128 minor frames that make up a major frame. A "minor frame" is one of the 128 main frames that occur during the subcom count from 0 through 127. A minor frame can be identified by the subcom count value. The only difference between the terms main frame and minor frame is that main frame refers to the 128 words in general and without regard to the subcom count value, while a minor frame refers to a specific main frame with a particular subcom count value. A "major frame" contains the words from all 128 minor frames as the subcom counter goes from 0 to 127, and requires 8 seconds for completion.

A frame sync interrupt is triggered as each new minor frame is received, thereby initiating processing of the previously received minor frame; this processing includes checking of upper and lower limits, scanning and stripping of selected experiment data, establishing command verification data, etc. The most recent data for each telemetry variable is taken from the minor frame and stored in a Latest Received Value (LRV) table, which is continuously accessible by both real-time and batch programs. Several flags are set by TMD to indicate, for example, that the latest full second of data is ready to be dumped onto the History Tape, or that data from a selected experiment is ready for processing.

3. Command Processing Section

The Command Processing Section (CMD) transmits one or more 64-bit commands to the experiments and spacecraft at a maximum rate of 60 milliseconds per command. A BCH cyclic check code, with generator polynomial

$$g(X) = (X^6 + X + 1)\ (X + 1)$$

is computed and stored in bits 58-64 of every command.

The CMD section checks the telemetry words TM(4) and TM(5) at every frame sync interrupt for validity using the BCH check code generated by

$$g(X) = (X^4 + X + 1)\ (X + 1)$$

The CMD downlink processor corrects single bit transmission errors, checks the command verification bits if command verification is requested, and checks the "flag bits" contained in the telemetry words TM(4) and TM(5).

Two modes of commanding are possible:

MODE 0 - Commands are issued with no verification.

MODE 1 - Commands are issued at the maximum rate until the output buffer is empty. Commands which the spacecraft fails to verify are retransmitted until the buffer is empty or the operator-specified number of retries have failed. If the latter occurs a message will be output to the CRT, Event Printer and the Spacecraft Event Printer informing the operator of the failure.

Each experiment or commandable spacecraft subsystem is associated with a "command program" which is executed whenever the system receives a sequence of command mnemonics related to that subsystem. The command programs are responsible for providing CMD with the proper bit patterns for minor or major mode commanding, operation code contents, and minor mode instructions. CMD can handle up to 64 commands at a time.

4. EDITOR Section

The EDITOR Section is used to create, modify and update automatic control sequences and procedures. The automatic control sequence, hereafter referred to as a "Schedule", contains a sequence of statements which may be commands to the experiments or spacecraft, or directives to the real-time programs of the AECS system. The Schedule may be used for experiment testing, checkout during spacecraft integration, or for directing in-flight operations. A "Procedure" is a short sequence of statements grouped together for convenience; a typical Procedure would be a set of directives or commands to turn on an experiment, or to turn it off if certain telemetry limits are exceeded.

5. Utility Section

The Utility Section (UTL) provides a powerful programming tool for the on-line debugging of real-time computer programs. UTL was conceived as a result of the schedule pressures placed on the AECS programming effort; the additional work in creating UTL saved considerable time in the later checkout stages of AECS applications programs.

Each of the above sections will be described in more detail on the following pages.

II. EXECUTIVE CONTROL SECTION - ECS

The ECS supervises all real-time processing operations. It receives control of the computer every 100 milliseconds through a clock interrupt and proceeds to scan a series of system flags associated with numerous processing tasks. Some of these tasks perform basic housekeeping functions, such as surveying and servicing the Test Conductors' Panel, connecting and arming all AECS interrupts, updating the History Tape, and driving the analog strip chart recorders. Other tasks initiated include interpreting all messages submitted to the system, loading and executing AECS overlay programs, and performing system utility functions.

1. ECS Hardware Configuration

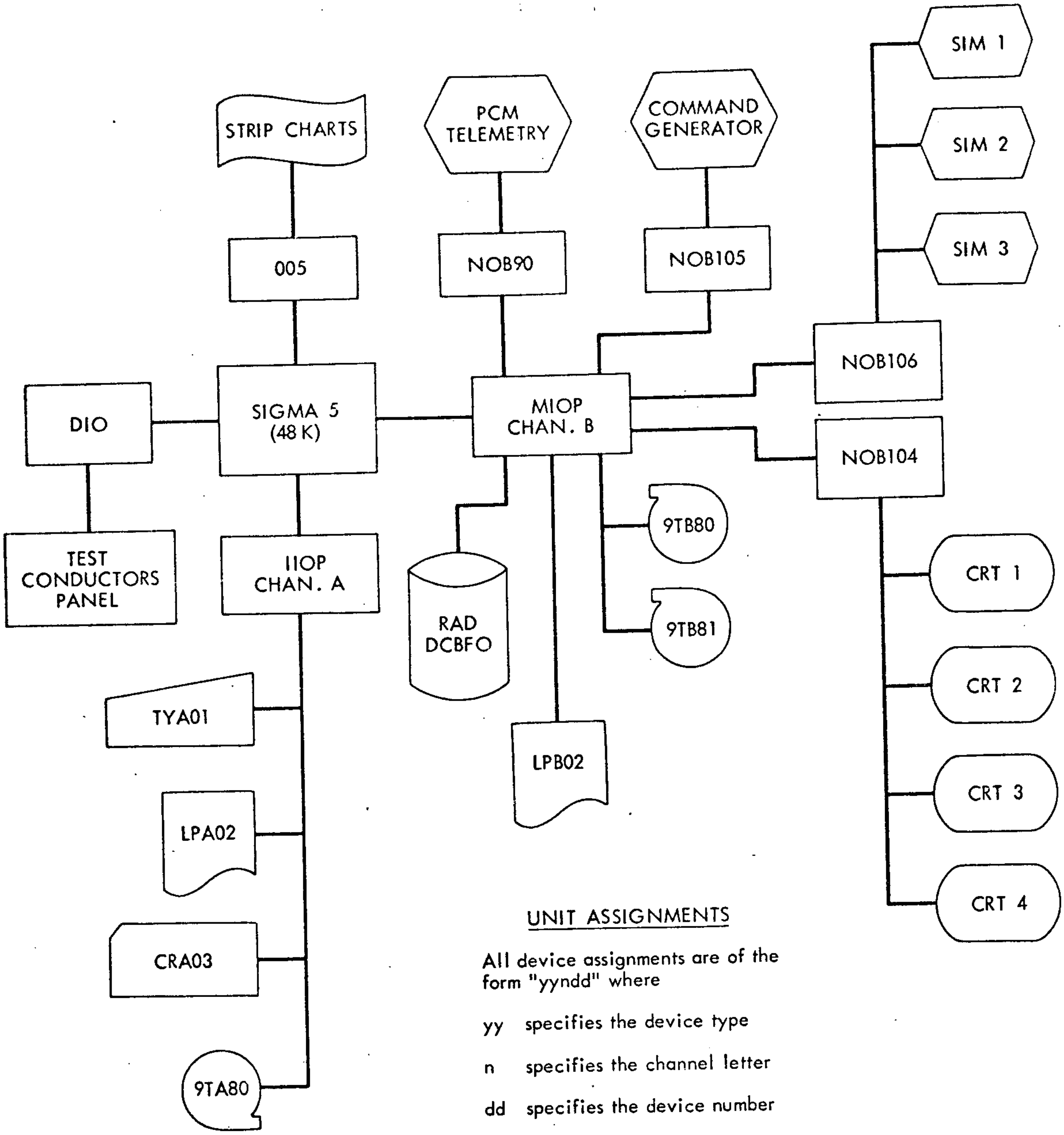
A diagram of the AECS hardware configuration is shown in Figure 1. ECS controls two line printers and two 9-track IBM-compatible tape drives. The line printer LPA02 is referred to as the Event Printer or Snapshot Printer, and has the operational labels EP and SN. The second line printer, LPB02, is referred to as the Spacecraft Event Printer and has the operational label of LO. A magnetic tape, either unit 9TB82 or 9TB83, is defined as the History Tape HT and is used for saving raw telemetry data for later off-line processing and analysis. Another tape, 9TA80, is defined as the Control Tape CT and is used for Schedule executions.

In addition, ECS controls several XDS System Keyboard Display cathode ray tubes (CRTs). These television screen/keyboards are general purpose interactive communications devices that allow input to and output from the system. All AECS operational labels are shown in Figure 2.

There are 16 interrupts available in the Sigma 5 computer, and most of these are used by ECS. The interrupt assignments are shown in Figure 3. The priority ordering scheme, while still evolving, has been quite satisfactory. AECS usage of the RAD files is shown in Figure 4.

2. ECS Message Interpreter

Subroutine MSGINTRP decodes and interprets all messages submitted to the AECS system. MSGINTRP immediately displays each 80-character input message simultaneously on line 19 of all CRT screens and also on the Event Printer. It separates, sequences, and stores up to 20 parameters in a buffer later used by the overlay programs. Then, depending on the type of message, it transfers control to the relevant message post-processor, in which the various ECS scan flags are set.



UNIT ASSIGNMENTS

All device assignments are of the form "yyndd" where

yy specifies the device type

n specifies the channel letter

dd specifies the device number

Thus, "LPB02" indicates the line printer on channel B, device #2.

Figure 1. Hardware Configuration for the AECS System

LABEL	UNIT	FUNCTION
HT	9TB82	History Tape
CT	9TA80	Control Tape for Schedules
EP	LPA02	Event Printer (800 1 pm) (RCA 1100 1 pm)
SN	LPA02	Snapshot Printer (800 1 pm) (RCA 1100 1 pm)
LP	LPB02	Line Printer (1100 1 pm)
C	TYA01	Sigma 5 Control Command Input
OC	TYA01	Operator's Console
LO	LPB02	Listing Output
LL	LPB02	Listing Log (RBM Job Cards)
DO	LPB02	Diagnostic Output (Dumps)
CO	9TB80	Compressed Output
CI	9TB81	Compressed Input
BO	9TB80	Binary Output
BI	CRA03	Card or Binary Input
SI	CRA03	Symbolic (Source) Input
SO	9TA80	Symbolic Output

Figure 2. RBM and AECS Operational Labels

All messages to AECS are identified by special symbols beginning in column 10 of the card image. All references to the simulators are applicable at T&E only. Use of these special symbols speeds up system reaction time by reducing the number of table look-up operations needed to convert the mnemonic string to the desired form for the post-processors. The symbols and corresponding message types are, in order of priority:

* - System Directive

/ - Subsystem Command (Real-Time)

- Spacecraft Command (Stored)

$ - Simulator Directive

INTERRUPT	DESCRIPTION
X'58'	Clock 1 counter (60 Hz), triggers ECS X'6B' every 100 milliseconds.
X'59'	Clock 2 counter (2000 Hz), used to generate simulated real-time telemetry and for the TMD frame sync window.
X'5A'	Clock 3 counter (8000 Hz) used to simulate timing between telemetry words for strip charts.
X'60'	I/O Endaction for command transmission through the 7601.
X'61'	I/O Endaction for transmission of spacecraft simulator data through the 7929.
X'62'	Telemetry Frame Sync Interrupt.
X'63'	—
X'64'	I/O Endaction for the spacecraft Event Printer.
X'65'	I/O Endaction for the Event Printer.
X'66'	I/O Endaction for the Snapshot Printer.
X'67'	I/O Endaction on telemetry History Tape.
X'68'	I/O Endaction for the CRTs.
X'6A'	I/O Endaction for RAD transmission of real-time telemetry data. Used by TMD.
X'6B'	ECS system flag scan. Main program of AECS.
X'6C'	ECS queue processing.
X'6D'	ECS Page execution.
X'6E'	ECS directive and command program execution.
X'6F'	RBM Control Task interrupt.
CAL2	Trap for CRT display instructions.
CAL3	Trap for Event Printer and Snapshot Printer operations.

Figure 3. AECS Interrupt and Trap Assignments

AREA	TRACKS	FUNCTION
SP	70	System programs - RBM, FORTRAN, MACRO-SYMBOL
FP	157	Foreground programs - AECS root and segments, directives, commands, snapshots, etc.
BP	75	Background program area
D1	30	Foreground data files - PEND, ACTV, STD, RAWFILE
D2	20	Background data files
D3	50	Procedures
XA	10	IOEX file
BT	100	Background temporary area - OV and GO files, scratch files X1, X2, , , X9.
512 Tracks Total		

Figure 4. RAD Storage Areas Assigned by AECS

? - Utility Directive

& - Procedure Call

A "directive" (*) is defined as a control instruction for the AECS which effects the state of the system; examples of directives are requests for page displays, starting or stopping real-time telemetry processing, entering a wait state, etc. A "command" (/) is an experiment-related overlay program which generates one or more 64-bit commands to be transmitted to the experiments. A "simulator directive" ($) is a program which generates one or more 32-bit words for the spacecraft simulators. A "utility directive" (?) is a subroutine which dynamically lists, displays, or modifies computer memory locations in order to assist in program debugging. A "procedure" (&) is group of combined directives and commands which are linked together as a convenience.

As soon as each message code is identified, control is passed to one of several post-processing routines:

* goes to DIRINPUT and DIRLOAD

/ goes to CMDINPUT and DIRLOAD

# goes to DPOUND

$ goes to SIMPROC.

? goes to UTILITY

& goes to PROCINP

3. ECS Flag Scan

A number of flags — locations in core — are periodically inspected by ECS to determine the status of real-time operations and initiate processing tasks. Flags currently examined by ECS are: (See Figure 5.)

NBREADY — if non-zero, indicates that the latest one second (16 minor frames) of PCM telemetry data is ready for transfer onto the History Tape HT. Byte 0 of NBREADY contains a pointer indicating which of 8 groups of 16 minor frames each is ready to be dumped, and halfword 1 of NBREADY contains the first word address of the data. This flag is set by TMD and cleared by HTDUMP.

TMDREADY — if non-zero, indicates that the TMD program must connect and arm the telemetry frame sync interrupt in order to begin receiving real-time telemetry data. This flag is set by the *START directive and reset by TMD.

CRTBUSY — if non-zero, indicates that the CRT displays are busy and not available for input or output operations at the current time. If zero, the CRTs are available, and each of the CRT keyboards are queried to find out if any new messages have been submitted to the system. This flag is set by the CRT handler and cleared by the ECS I/O endaction routine.

PROCFLAG — if byte 0 is non-zero, indicates that the next statement of a currently executing procedure has been read into core and is ready for operation. Byte 0 is actually the RBM I/O completion code set by the CAL1 READ operation and reset by the PROC processor.

SCHDFLAG — if non-zero, indicates that the schedule processor is in operation: that is, messages are being read in automatically and continuously from the schedule tape. This flag is set by the *SCHEDULE directive and reset by an end-of-data condition on the schedule tape.

MSGFLAG — if non-zero, indicates that a new message has been submitted to the AECS system from some input device. Byte 0 contains the identification of the calling unit, and halfword 1 contains the address of the message buffer.

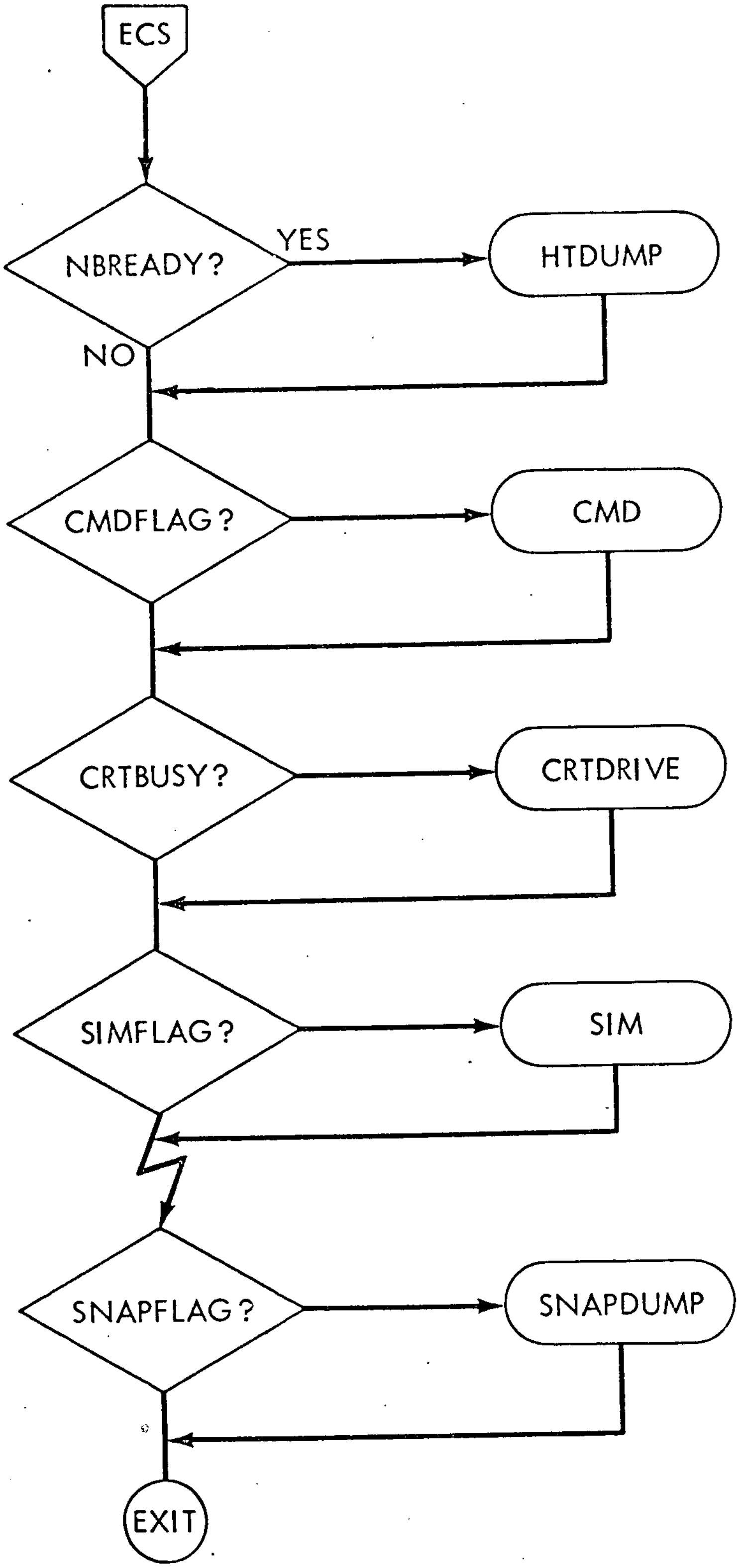


Figure 5. Basic Flow of the Executive Control System (ECS)

SIMFLAG — if non-zero, indicates that a simulator directive request has been received and needs processing. SIMFLAG is set by MSGINTRP and cleared when the desired program has completed execution. SIMFLAG contains the mnemonic of the simulator directive.

CMDFLAG — if non-zero, indicates that a command program has been requested. CMDFLAG is set by MSGINTRP and cleared when the command program is finished. CMDFLAG contains the 4-character experiment mnemonic.

DIRFLAG — if non-zero, contains the 8-character mnemonic of the directive program when has been requested. DIRFLAG is set by MSGINTRP and cleared after the directive program has finished.

PAGEFLAG — if non-zero, indicates that one or more CRT display pages must be called in (overlaid) and executed in order to update the display. PAGEFLAG is set and cleared by the *PAGE directive.

UTLFLAG — if non-zero, contains the mnemonic of the desired Utility directive. UTLFLAG is set by MSGINTRP and cleared after the desired utility program has been initiated.

UPDATE — if non-zero, indicates that the currently executing directive or command program will remain in core and be re-executed at one second intervals, unless another message request is received or the program is cancelled. UPDATE is set by the individual programs and cleared by MSGINTRP or CANCEL.

CRFLAG — if non-zero, indicates that messages are expected from the card reader, defined as Unit B in AECS. CRFLAG is set by setting Sense Switch 1 on the computer console, or by depressing the "CARD" button on the Test Conductors' Panel. CRFLAG is cleared when the sense switch is reset or when there are no more cards in the hopper.

TTYFLAG — if non-zero, indicates that messages are expected from the computer teletype console (AECS Unit A). TTYFLAG is set by setting Sense Switch 2 on the computer console and cleared when the (one) message is transmitted by the End-of-Message (EOM) or New Line (NL) key on the teletype.

PANMSG — If nonzero, indicates that input from the AE Test Conductor's Console has been received and is being processed. DIRLOAD sets the flag to -1 when the specified request is initialized. The flag is cleared after execution of the selected request is complete.

GRPEXEC — If nonzero, indicates that a Matrix or Group Command is being processed.

CMDERROR — If nonzero, it contains the address of a message to be displayed passed to ECS by CMD.

SOS:FLAG — If nonzero, contains the address of the call to the out-of-limits Procedure. ECS checkpoints the current Procedure, runs the out-of-limits Procedure. The interrupted Procedure or Schedule is given control after the out-of-limits Procedure is complete.

4. CRT Displays and Keyboards

ECS controls four XEROX Data Systems "System Keyboard Display" cathode ray tubes (CRTs). These television screen/keyboards, which are identified as CRT#1, CRT#2, CRT#3, and CRT#4, are general purpose interactive communications devices which allow input to and output from the system. Each CRT screen displays alphanumeric characters on a matrix 20 lines down by 80 columns across. The top line is defined as line #1 and the leftmost column is column #1. Character transmission occurs at the rate of 1200 characters per second.

Messages are typed at the CRT keyboards in LOCAL mode. Figure 6 shows the keyboard arrangement. Transmission of messages is performed by depressing the SEND key. Errors in typing may be eliminated by backspacing and typing over erroneous characters, by depressing the DELETE key which removes the character under the cursor, or by depressing the LINE ERASE key which erases the entire line to the right of the cursor. The updating of a CRT screen can be temporarily halted by depressing any of the four FUNCTION keys F1 - F4. Updating will be restarted after 45 seconds or after the SEND key is depressed.

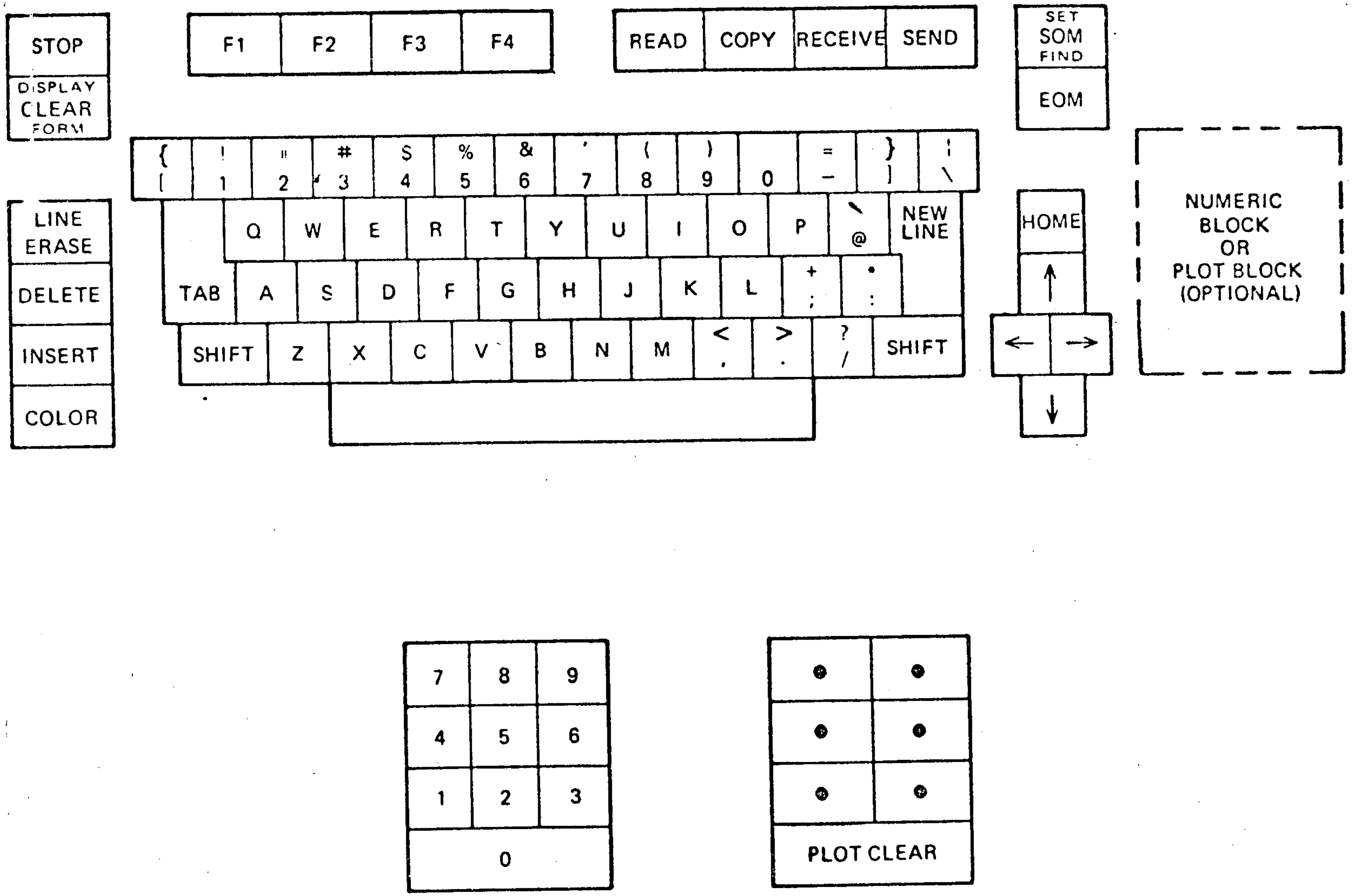
The bottom three lines of each display screen are reserved by AECS for the following purposes:

Line 18 - System error messages

Line 19 - Statement which is currently executing

Line 20 - Next statement to be executed.

Each CRT screen, therefore, is limited to displaying no more than 17 lines of user-oriented data.



NUMERIC BLOCK

PLOT BLOCK

Figure 6. System Keyboard on Each CRT Device

While each CRT operator can issue commands to the spacecraft and experiments, he may be enabled or disabled from doing so by the *ENABLE and *DISABLE directives which can be submitted only through CRT #1. CRT #1 thus enjoys executive control over all other CRTs.

5. Pages

A "Page" is an overlay segment program which displays real-time telemetry data in raw or processed form on any specified CRT screen. Each Page program receives control at regular time intervals (every 3 seconds), and operates autonomously in the system; that is, there is no restriction on its making use of the full range of CRT display capabilities, such as blinking, alarming, and plotting. When a Page program is initiated, it generates title and header information on the screens, then converts and displays the desired telemetry data; subsequent executions of the same Page program can bypass the transmission of unchanged title data and proceed directly with updating the telemetry data. Pages can operate in core simultaneously with command and directive programs.

6. Stripcharts

ECS services any number of analog strip chart pen records and event markers. These pens operate independently and simultaneously, with plotting speed and scale regulated by controls at the recorder console. Access to these pens is through the directives *CHARTON, *CHARTOFF, *ASSIGN, *SUPGRAF, and *SUPGFOF.

7. Snapshots

A "snapshot" is an overlay segment program similar to a directive or command, which produces a fixed format printout of telemetry or other data associated with the spacecraft, experiments, or simulators. The Snapshot Printer SN is used exclusively as the output device.

8. Procedure or "PROC"

A "Procedure" is a logically related set of system statements (directives, commands, snapshots, etc.) which are grouped together for compactness and operational efficiency. A Procedure call is a single statement resulting in the execution of all statement contained in that Procedure. Procedures are created by the AECS EDITOR program and placed on the RAD for rapid access. All Procedure calls are indicated by the prefix symbol &.

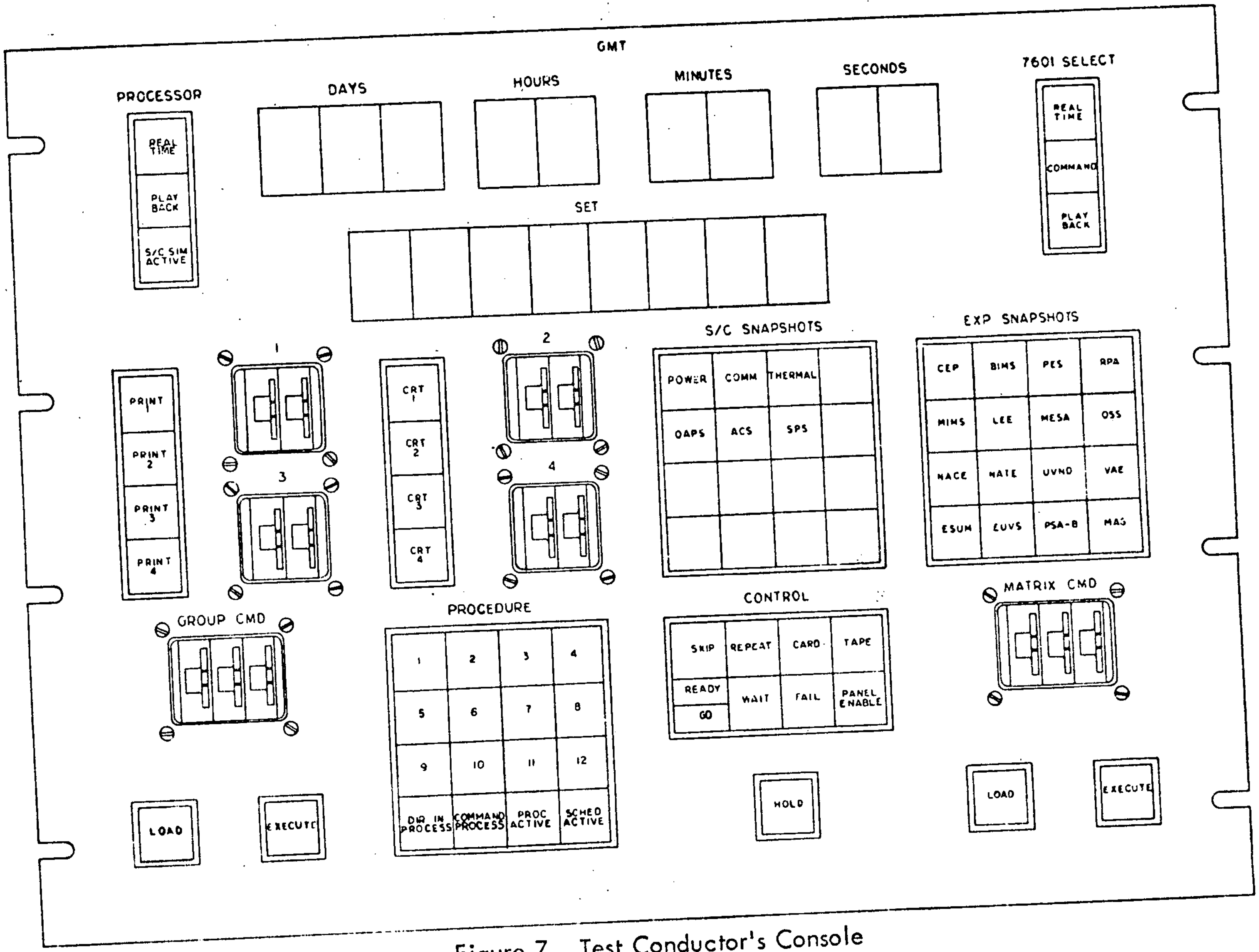


Figure 7. Test Conductor's Console

9. Schedule

A "Schedule" is an automatic control sequence. It can contain up to 10,000 directives, commands, procedures, snapshots, simulator directives, etc., which are executed either sequentially or as directed by the statements themselves. A Schedule is actually a magnetic tape created from an input card deck by the AECS EDITOR program. One of the functions of EDITOR is to add, delete, or modify existing schedules, as well as provide a listing of the schedule tape contents, including a full expansion of all imbedded Procedures. A Schedule is called for execution by the *SCHEDULE directive.

10. Test Conductors Console

A push-button console is provided to simplify the system operator's task of submitting requests to the system. The console allows requests for CRT Pages, snapshots, Group commands, Matrix commands, etc., and also displays the Greenwich Mean Time (GMT) and Spacecraft Elapsed Time (SET) in digital form. ECS acknowledges each request by illuminating the depressed push-button. Only one push-button operation can occur at a time.

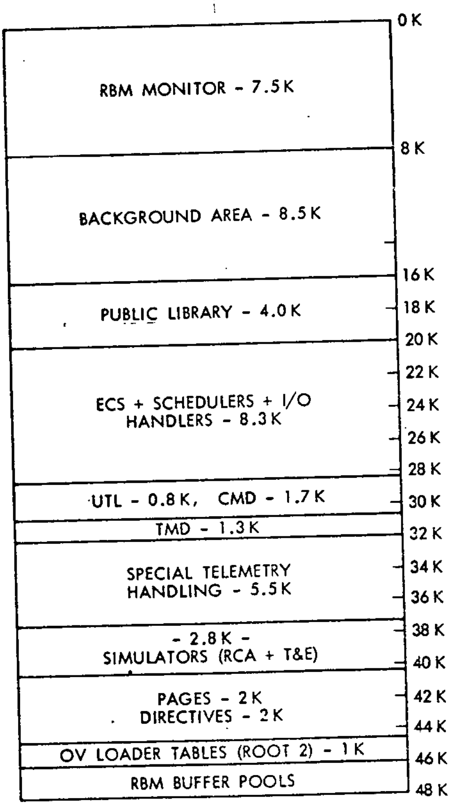


Figure 8. Core Layout for the AECS System

III. TELEMETRY DECOMMUTATION SECTION (TMD)

TMD receives and processes narrowband PCM telemetry data at the minor frame rate of 128 8-bit words (bytes) every 62.5 milliseconds. Each minor frame is checked for frame synchronization (coded as X 'FAF320' in words 1, 2 and 3) and specified words are limit-checked against preset limits (*LIMITS directive) to establish a gross estimate of experiment and spacecraft performance. If commands are concurrently being issued to the experiments or spacecraft, the command verification information from words 4 and 5 is extracted for use by the Command Processing Section. A specified number of telemetry words may be stripped out of each minor frame and saved for real-time use by data processing programs. Telemetry housekeeping is performed, and the relevant flags, pointers, counters, etc., are updated to reflect the current state of the system. Updated status can then be displayed in hard copy on the Snapshot Printer by "snapshot" programs, or dynamically on the CRTs through Page routines. Raw telemetry data is saved on the History Tape in 560-word records containing one second of telemetry data and the latest one second's worth of verified commands. (See Figure 9.)

1. Telemetry Notation

Telemetry words may be referenced in AECS through the following notation convention:

TM(I) or TM(I,J)

where

I = the telemetry word index, from 1 through 128

and

J = the subcom step number, also from 1 through 128. (See Figures 10 & 11.)

Individual programs requiring this notational form for input are responsible for checking the values of the subscripts to avoid erroneous references. The more sophisticated notational scheme described in Appendix F of this document has not been implemented in the AECS software.

2. Ping-Pong Buffers

There are two ping-pong buffers in TMD. As soon as the first ping-pong buffer is full, the second begins to receive data. These buffers are accessed

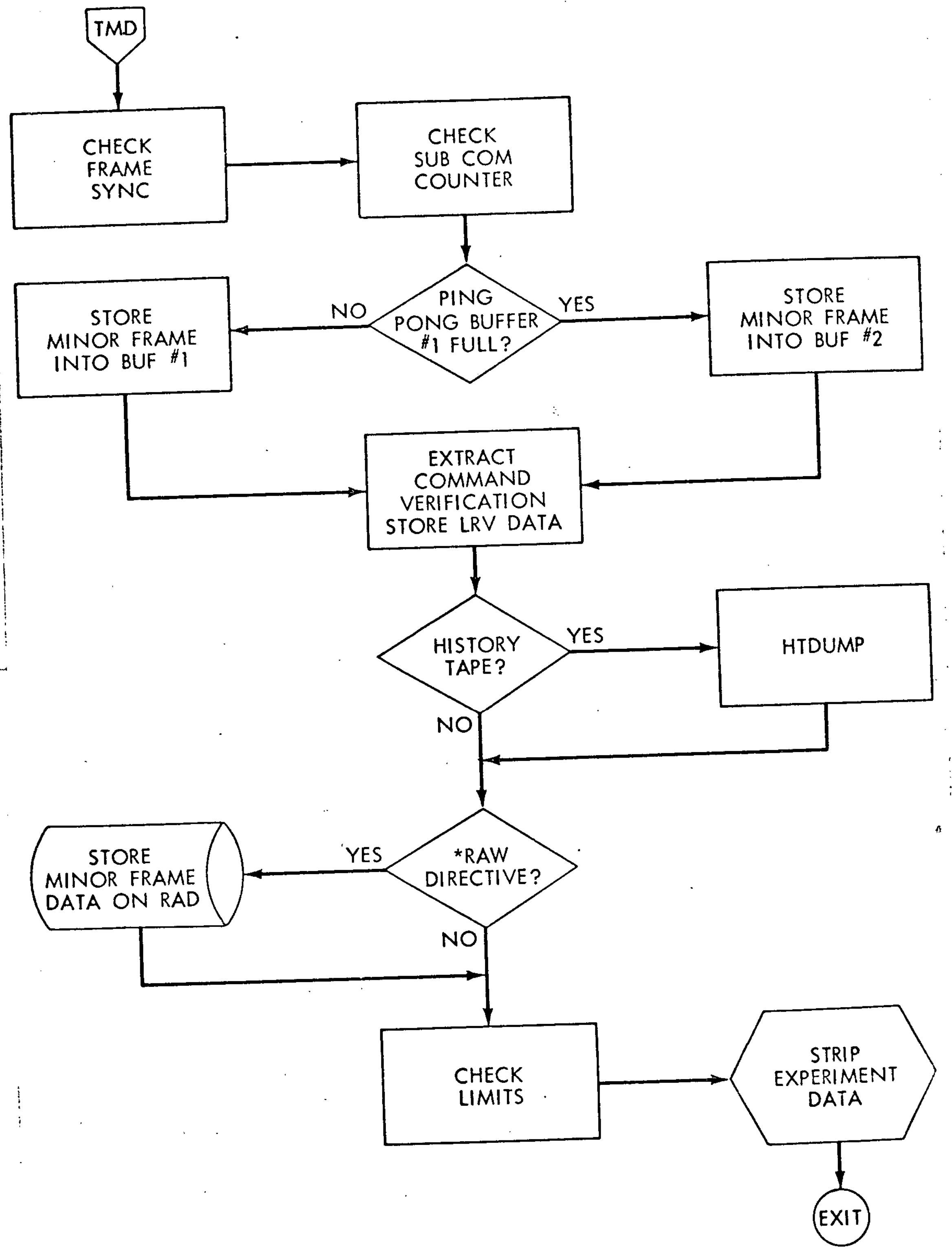

TMD
CHECK FRAME SYNC
CHECK SUB COM COUNTER
PING PONG BUFFER #1 FULL?
NO
YES
STORE MINOR FRAME INTO BUF #1
STORE MINOR FRAME INTO BUF #2
EXTRACT COMMAND VERIFICATION STORE LRV DATA
HISTORY TAPE?
YES
HTDUMP
NO
*RAW DIRECTIVE?
YES
STORE MINOR FRAME DATA ON RAD
NO
CHECK LIMITS
STRIP EXPERIMENT DATA
EXIT

Figure 9. Basic Flow of TMD Program

CHANNEL	SUB COM LEVEL
TM(11,J)	J = 1 to 8
TM(12,J)	J = 1 to 8
TM(13,J)	J = 1 to 8
TM(14,J)	J = 1 to 8
TM(17,J)	J = 1 to 4
TM(18,J)	J = 1 to 4
TM(20,J)	J = 1 to 8
TM(34,J)	J = 1 to 4
TM(35,J)	J = 1 to 4
TM(46,J)	J = 1 to 2
TM(47,J)	J = 1 to 2
TM(48,J)	J = 1 to 2
TM(55,J)	J = 1 to 4
TM(65,J)	J = 1 to 64
TM(66,J)	J = 1 to 128
TM(67,J)	J = 1 to 64
TM(68,J)	J = 1 to 128
TM(98,J)	J = 1 to 4
TM(99,J)	J = 1 to 4
TM(110,J)	J = 1 to 2
TM(111,J)	J = 1 to 2
TM(112,J)	J = 1 to 2
TM(119,J)	J = 1 to 4
TM(120,J)	J = 1 to 4
TM(128,J)	J = 1 to 16

Figure 10. Main Frame Sub-Commutated Channels

each time the frame sync interrupt (X'62') fires, at which time TMD executes a direct read sequence to bring the next 128 bytes into the appropriate slot in core. A "window" is provided to verify that the frame sync interrupt does indeed fire at the prescribed rate of 62.5 milliseconds; if it fires before 60 milliseconds has elapsed, or after 67 milliseconds has elapsed, a message appears on the CRTs and the Event Printer. The next minor frame is stored in the location specified by the contents of the sub-com counter, TM(37).

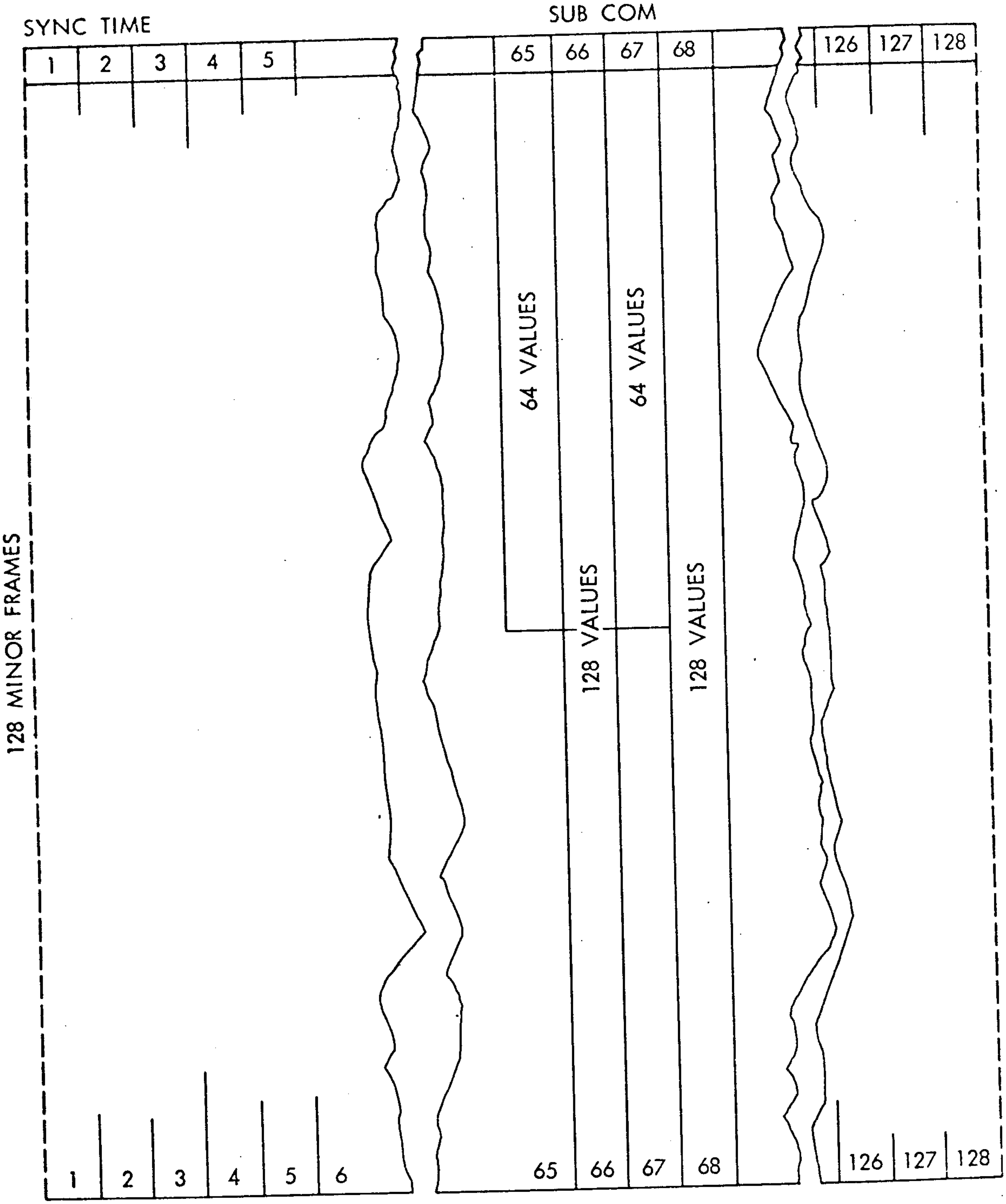


Figure 11. Complete Major Frame Map

3. Latest Received Value Table (~~LVR~~ LRV)

TMD maintains an LRV table which contains the most recently received telemetry words, and which is accessible through LRVINDEX by both real-time and background programs for data processing. The LRV table actually consists of two arrays, LRVMINOR and LRVSUBCM, corresponding respectively to 32 full minor frames and to all 128 steps of sub com channels 65, 66, 67, and 68. LRVMINOR is 1024 words long, and is updated every 62.5 milliseconds; LRVSUBCM is 512 bytes long, and similarly updated. (See Figure 12.)

4. Limit Checking

The capability exists for TMD to check each telemetry word against specified high and low limits, and to execute a Procedure if the limits are exceeded two times in succession. This Procedure will interrupt any active Schedule or Procedure and return control to the Schedule or Procedure upon completion. If the limits for a telemetry word are exceeded five times in succession, printing of the out-of-limit condition is terminated for that particular telemetry word, but the limit checking continues. See *LIMITS, *LIMOUT, *LIMON, *LIMOFF, and *LIMSET directives for details.

5. Conversion

While TMD does not itself convert raw telemetry data into engineering units, the capability is provided for other real-time or data processing programs to do so. TMD maintains the coefficient table (TMDCOEF) which is built up by the *CONVCOEF directive. The contents of TMDCOEF are the polynomial coefficients A0, A1, A2,,,. . . ., AN, which form the equation

$$Y = A0 + X* (A1 + X* (A2 + X* (A3 + \ldots X* AN))))))$$

6. History Tape Recording

All PCM narrowband telemetry data can be recorded on magnetic tape. If the *HTSTART directive has been issued, TMD will dump telemetry data every second onto 9TB82 or 9TB83 through the HTDUMP program. HTDUMP is activated by the NBREADY system flag; this program first checks to insure that the History Tape is ready to receive data, and then it proceeds to execute the I/O instructions. The actual format of the 560-word History Tape record is shown in the explanation of the *HTSTART directive. This particular record size was chosen in view of the size of the telemetry ping-pong buffer, the possibility of tape channel lock-out during I/O, the readability of 560-word records by background programs, and the compatibility of this size with other computer systems.

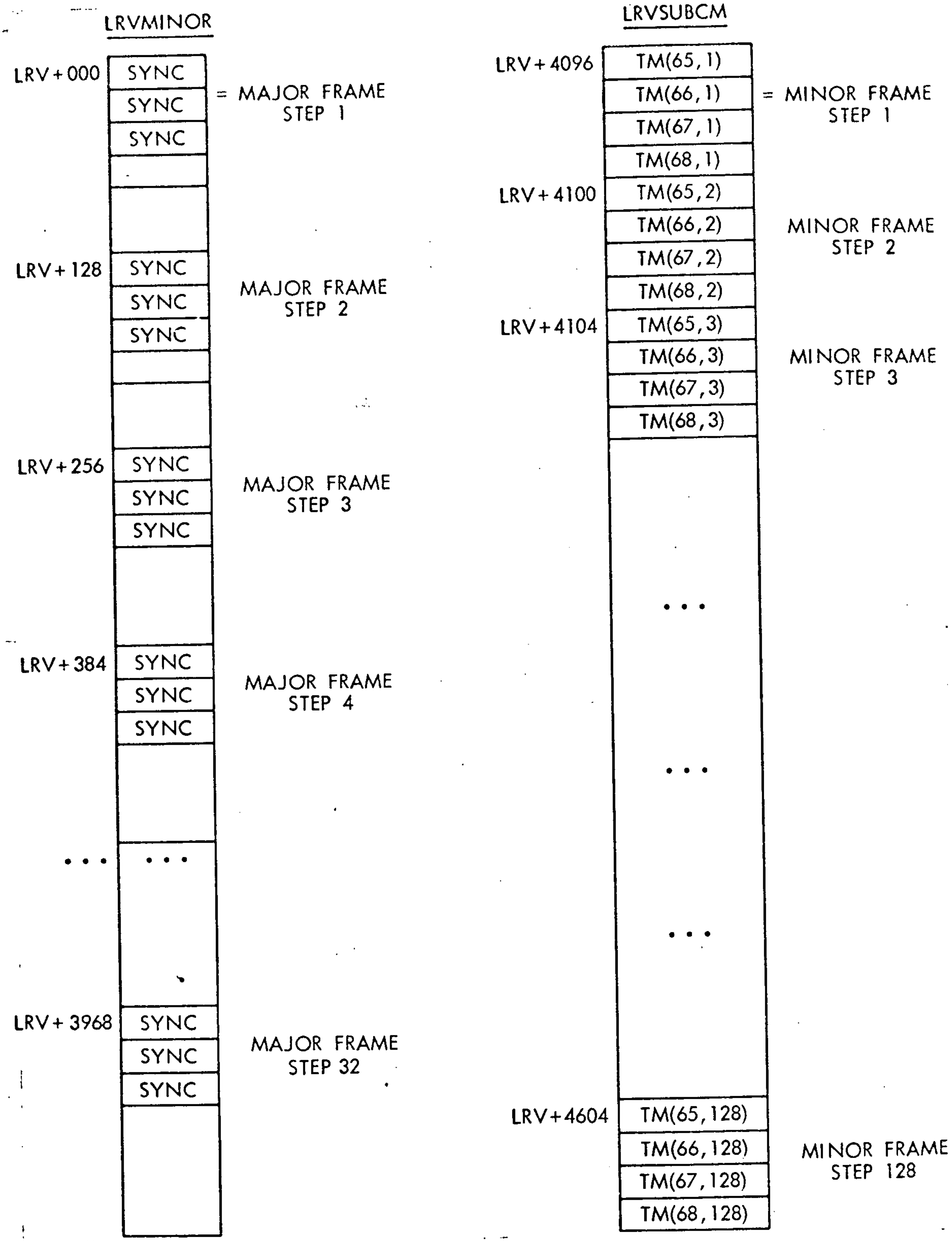


Figure 12. Latest Recorded Value (LRV) Table. Byte Addressing

It should be noted that provision has been made, through the *HTSTOP directive, to stop recording data at any time in order to conserve tape usage and system overhead. In addition, if one of the History Tapes is nearing end-of-reel, another tape may be readied and the *HTSTART directive re-issued with reference to the new tape; in this way, no data is lost during important recording sessions.

Each 560-word record occupies 2.7 inches on tape (at 800 bpi). The inter-record gap is approximately 0.65 inches. Each record dump thus takes 3.35 inches of tape. For a full 2400-foot reel, over 8,700 records can be dumped, or about 2.4 hours worth of data. (See Figure 13.)

7. Command Memory Dump

Certain telemetry words in the minor frame are stripped out by TMD and stored in the array CMEMDUMP. These words correspond to the command memory dump transmission down the 7.2KB line. The 46 words are allocated as follows:

1.	TM(1)	SYNC	2.	TM(2)	SYNC	3.	TM(3)	SYNC
4.	TM(4)	STATUS	5.	TM(5)	STATUS	6.	TM(7)	UVNO
7.	TM(8)	UVNO	8.	TM(15)	RPA	9.	TM(16)	BIMS
10.	TM(17)	S/C	11.	TM(18)	S/C	12.	TM(19)	MIMS
13.	TM(20)	EUVS	14.	TM(27)	NATE	15.	TM(28)	NATE
16.	TM(37)	SUBCOM	17.	TM(39)	PES	18.	TM(40)	PES
19.	TM(47)	VAE	20.	TM(48)	VAE	21.	TM(51)	RPA
22.	TM(52)	RPA	23.	TM(59)	NATE	24.	TM(60)	NATE
25.	TM(65)	SUBCOM	26.	TM(66)	SUBCOM	27.	TM(67)	SUBCOM
28.	TM(68)	SUBCOM	29.	TM(71)	TAL	30.	TM(72)	UVNO
31.	TM(79)	RPA	32.	TM(80)	BIMS	33.	TM(81)	S/C
34.	TM(82)	S/C	35.	TM(83)	MIMS	36.	TM(84)	MIMS
37.	TM(91)	NATE	38.	TM(92)	NATE	39.	TM(103)	PES
40.	TM(104)	PES	41.	TM(111)	VAE	42.	TM(112)	VAE
43.	TM(115)	RPA	44.	TM(116)	NACE	45.	TM(123)	NATE
46.	TM(124)	NATE						

Words 1 - 32	Minor Frame I I = 1,17,33,..,113
Words 33 - 64	Minor Frame I + 1
Words 65 - 96	Minor Frame I + 2
Words 97 - 128	Minor Frame I + 3
Words 129 - 160	Minor Frame I + 4
Words 161 - 192	Minor Frame I + 5
Words 193 - 224	Minor Frame I + 6
Words 225 - 256	Minor Frame I + 7
Words 257 - 288	Minor Frame I + 8
Words 289 - 320	Minor Frame I + 9
Words 321 - 352	Minor Frame I + 10
Words 353 - 384	Minor Frame I + 11
Words 385 - 416	Minor Frame I + 12
Words 417 - 448	Minor Frame I + 13
Words 449 - 480	Minor Frame I + 14
Words 481 - 512	Minor Frame I + 15
Words 513 - 516	GMT Time in EBCDIC - GMT DDD:HH:MM:SS
Words 517 - 520	Unassigned

Words 521 - 524	First command verified in the past 1 second (16 minor frames) of the form NNNNTTTTCCCCCCCC, where
	NNNN = Experiment mnemonic
	TTTT = GMT time, 32-bit binary
	CCCCCCCC = 64-bit command

Words 525 - 528	Second Command
Words 529 - 532	Third Command
Words 533 - 536	Fourth Command
Words 537 - 540	Fifth Command
Words 541 - 544	Sixth Command
Words 545 - 548	Seventh Command
Words 549 - 552	Eighth Command
Words 553 - 556	Ninth Command
Words 557 - 560	Tenth Command verified in the past second

Figure 13. Format of Each Record on the History Tape

IV. COMMAND PROCESSING SECTION - CMD

CMD generates one or more commands from command mnemonics and controls their transmission to the spacecraft or experiments. Commands may be transmitted singly or in groups, and in one of ~~three~~ two verification modes. Commands can be submitted to the system through the card reader, teletype, RAD (via a Procedure), schedule tape, or individually through any of the CRT keyboards-providing the appropriate *ENABLE or *DISABLE statements have been issued by the system operator at CRT #1. (If a command is identified as being "dangerous", a message is displayed to the system operator, and the system enters a "HOLD" state during which time no new processing tasks can be initiated; no dangerous commands are sent without operator approval, which is furnished by means of the ~~*GO~~ or ~~*DELETE~~ directive.) (See Figure 14.)

/CONTINUE, /DELETE

1. Verification Modes

Two modes of commanding are possible:

MODE 0 - Commands are issued with no verification.

MODE 1 - Commands are issued at the maximum rate until the output buffer is empty. Commands which the spacecraft fails to verify are retransmitted until the buffer is empty or the operator-specified number of retries have failed. If the latter occurs a message will be output to the CRT, Event Printer and the Spacecraft Event Printer informing the operator of the failure.

2. Commanding Sequence

When a command statement (beginning with the symbol /) is submitted to AECS, the following sequence occurs:

1. MSGINTRP separates the parameters in the command mnemonic string.
2. DIRLOAD loads the specified command decoding program, providing that no other program is currently executing.
3. The command program beings execution and builds portions of the final command through a series of table look-up operations.
4. The command program transfers control to COMMAND for final processing, transmission, and verification.

When COMMAND receives control, it performs the following steps:

1. Checks the validity of the command mnemonic in the case of experiment commands.

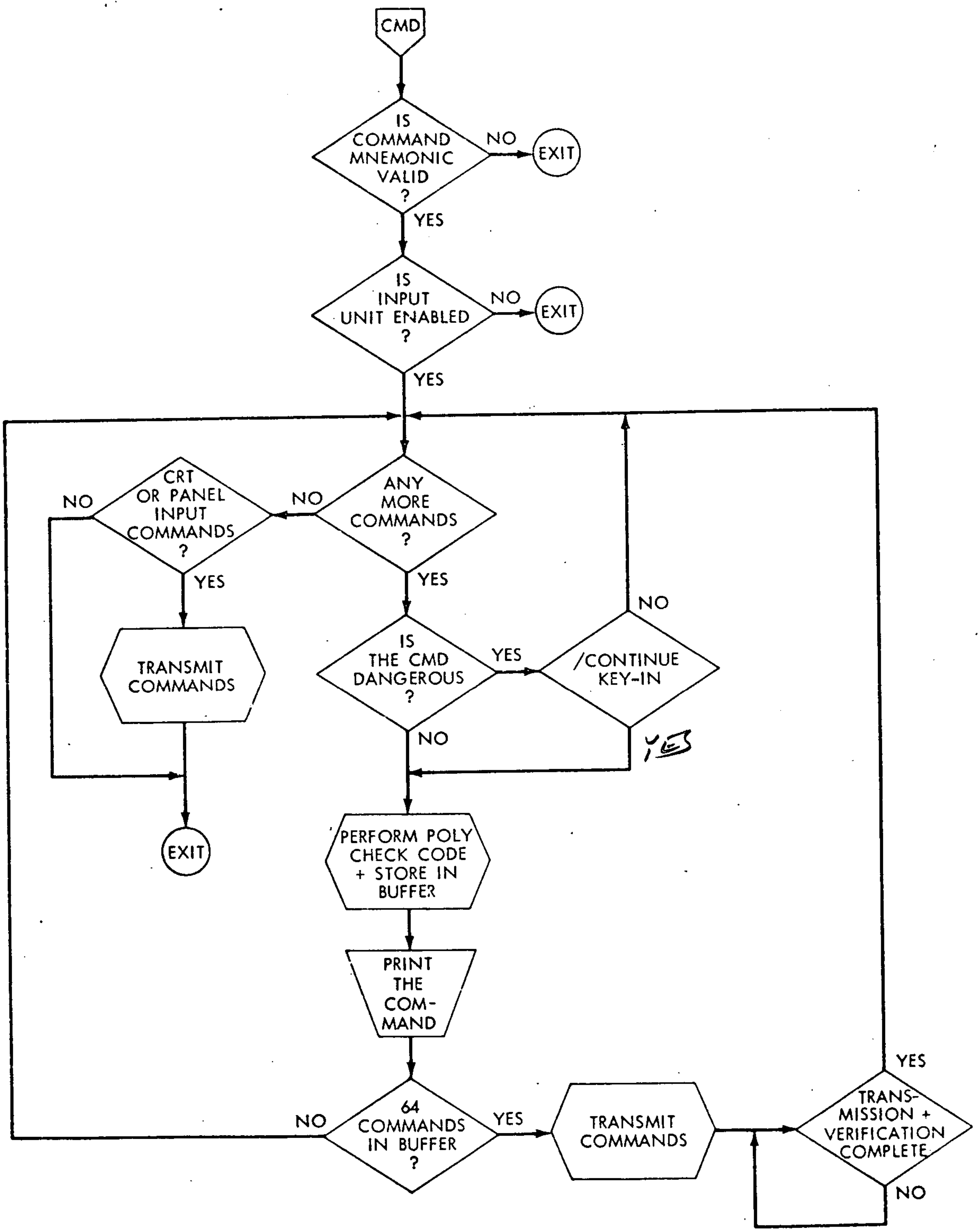


Figure 14A. CMD Transmission Loop

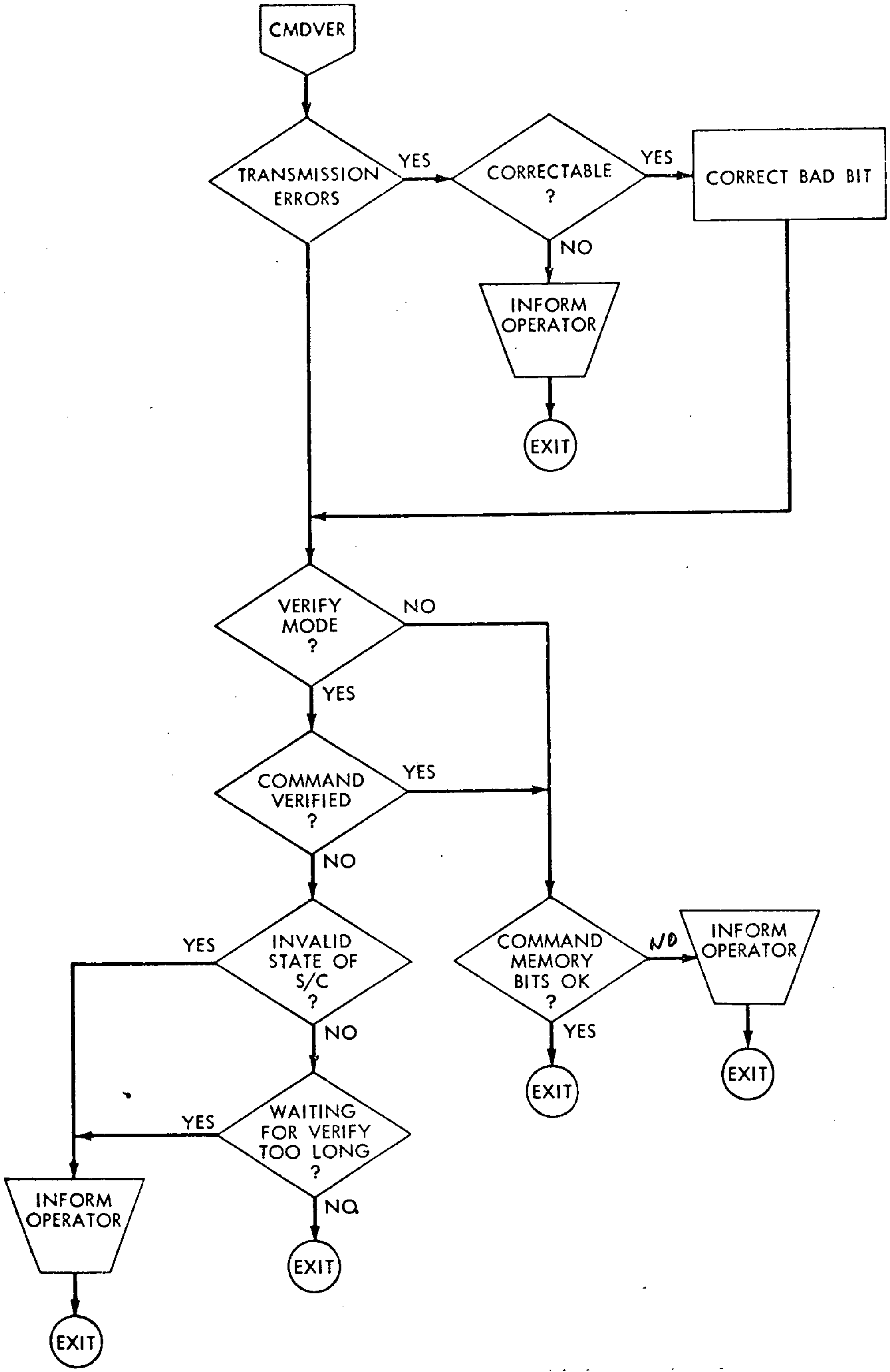


Figure 14B. CMD Downlink Loop

2. Checks the eligibility of the device issuing the command. If an *ENABLE directive was not issued to allow the unit to generate commands, an error message is displayed and the command is ignored; control returns to the command decoding program.

3. Picks up all of the commands (up to 64) to be sent and transfers them to the command output buffer.

4. Checks the number of transmissions allowed and the command verification mode. Both parameters are specified by the *VERIFY directive.

5. Constructs and inserts the cyclic check code bits into the 64-bit command word, ~~then transmits the command~~. The cyclic check code is a BCH code whose generator polynomial is

$$g(X) = (X^6 + X + 1)\ (X + 1)$$

6. If the unit requesting transmission of a command was a CRT or the AE Test Conductor's Console, the commands are transmitted immediately. Otherwise, the commands are stored in the command buffer until a non-command input request is encountered by the system or 64 commands have been stored in the command buffer. Actual I/O Transmission of a string of one or more 64-bit commands requires preceding the string by a 64-bit "preamble" for bit synchronization. This preamble contains 63 zeros followed by a one. Control is returned to the command decoding program after the 64-bit commands has been placed in the I/O buffer.

7. If command verification was requested, at every Frame sync interrupt the command verification words TM(4) and TM(5) are interrogated for command verification. These words are continuously checked for errors in transmission, and command memory flag bit errors.

3. Command Programs

A command program is an overlay program within AECS designed to interpret the English-language mnemonics established for an experiment and construct the basic fields of the 64-bit command. Each experiment is uniquely associated with a command program. The mnemonics are used to simplify the commanding process, and correspond to one or more bit fields in the command word. When these fields have been constructed, the command program transfers control to CMD, which then transmits the command.

A typical command has the form

/NNNN,P1,P2,P3,,,,,.

where

NNNN - is the experiment name (3 or 4 characters)

Pi - are mnemonics which are to be interpreted.

4. Command Word Format

The actual format of the 64-bit command word is as follows:

Bits 1-7 - X'EO' - Satellite address code

Bit 8 - Spare bit = 0

Bits 9, 10 - Decoder select code; 01 = decoder 1, 10 = decoder 2 and 00 or 11 signified neither decoder

Bits 11, 12 - Data bus control code; 00,01,10 - no minor mode data, 11 signifies minor mode data is present

Bit 13 - Spare = 0

Bit 14 - Spare = 0

Bit 15 - Spare = 0

Bits 16-24 - 9 bit OP CODE

Bits 25-56 - 32 bit MINOR MODE DATA; these bits are all 0's if minor mode data is not included

Bit 57 - Spare = 0

Bits 58-64 - Cyclic code check bits

It should be noted that for all command transmissions, whether for individual 64-bit commands, or a group of such commands, the command string is always preceeded by a special "frame sync" word containing 63 logic 0 bits followed by a logic 1 bit. (See Figure 15 for Command Word Structure.)

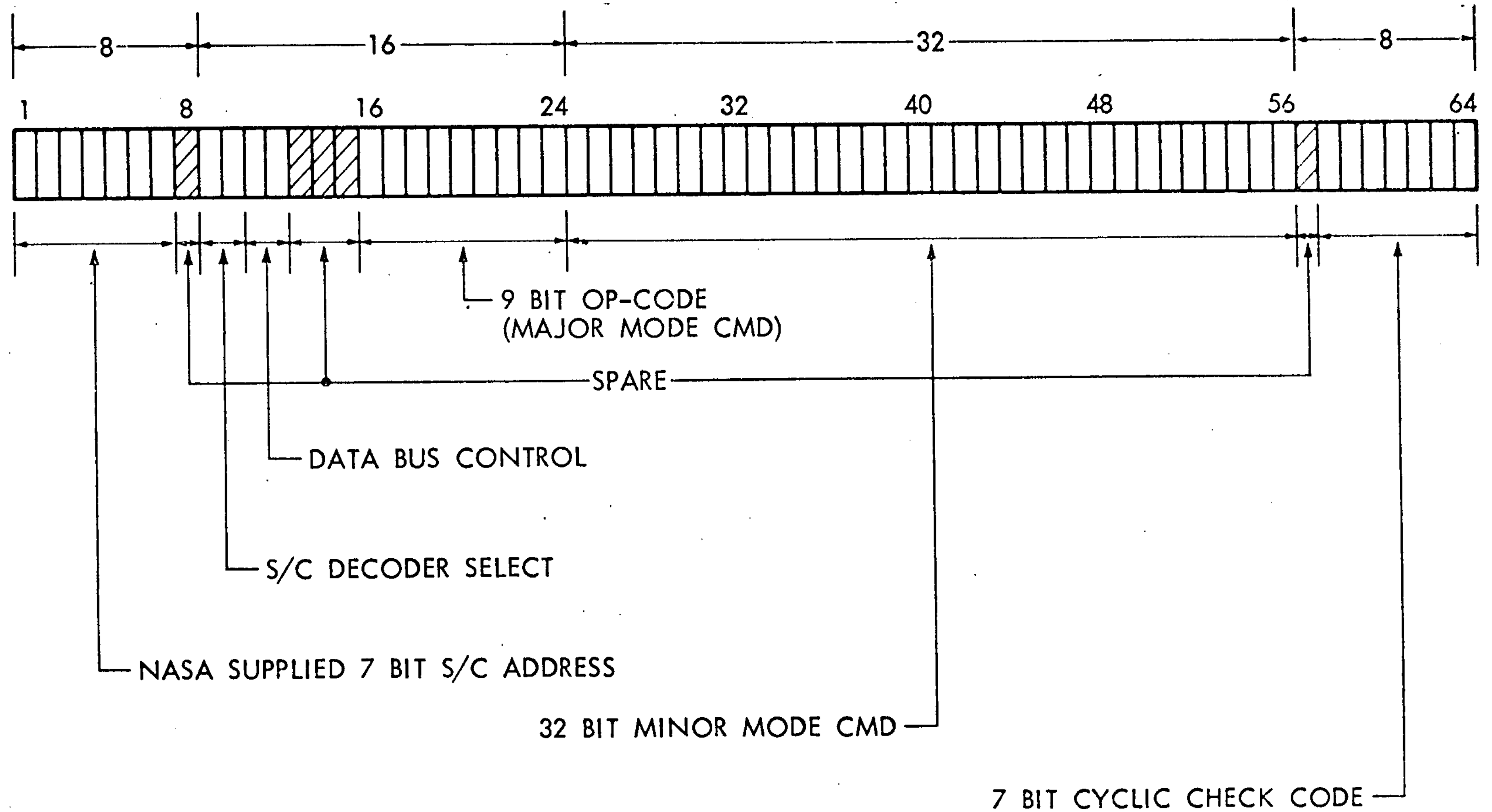


Figure 15. Command Word Structure - Real-Time Command

V. EDITOR

EDITOR is a real-time program that creates Procedures and Schedules. A Schedule is a chain of directives, commands, and Procedure calls combined together, sequenced, and given a name for later reference. Procedures are groups of commands and directives that are related to a specific function. A Schedule is put onto a magnetic tape, Procedures onto the RAD (D3 area). Procedures are also referred to as "PROCS".

To clarify the difference between a Schedule and a Procedure, an analogy can be drawn between EDITOR and the FORTRAN compiler; a schedule corresponds to a main program, a procedure to a subroutine, and the directives and commands to individual instructions.

Input to EDITOR consists of (1) EDITOR Control Cards which determine the activity of EDITOR, and (2) Procedure and schedule statements; in the case of Procedures, the statements are directives and commands, and for a Schedule, the statements are directives, commands, and Procedures as well. (See Figure 16.)

1. Procedure Statements

Procedure statements have the following general format:

```
S  NNNNNNNN,PARM1,PARM2,,,,,,,.  COMMENTS
```

where

S - The symbol denoting a directive (*), a command (/), a stored spacecraft command (#), a utility function (?), a simulator directive ($), or a Procedure call (&). The symbol must be in column 10.

NNNNNNNN - The name of the directive, command, etc., being called. In the case of a Procedure call, NNNNNNNNN is the name of the Procedure.

PARMi - parameters

COMMENTS - descriptive or otherwise useful information which will be displayed on the CRTs and Event Printer.

Columns 1 through 9 and 73 through 80 are used by EDITOR and must be left blank, or the contents will be ignored.

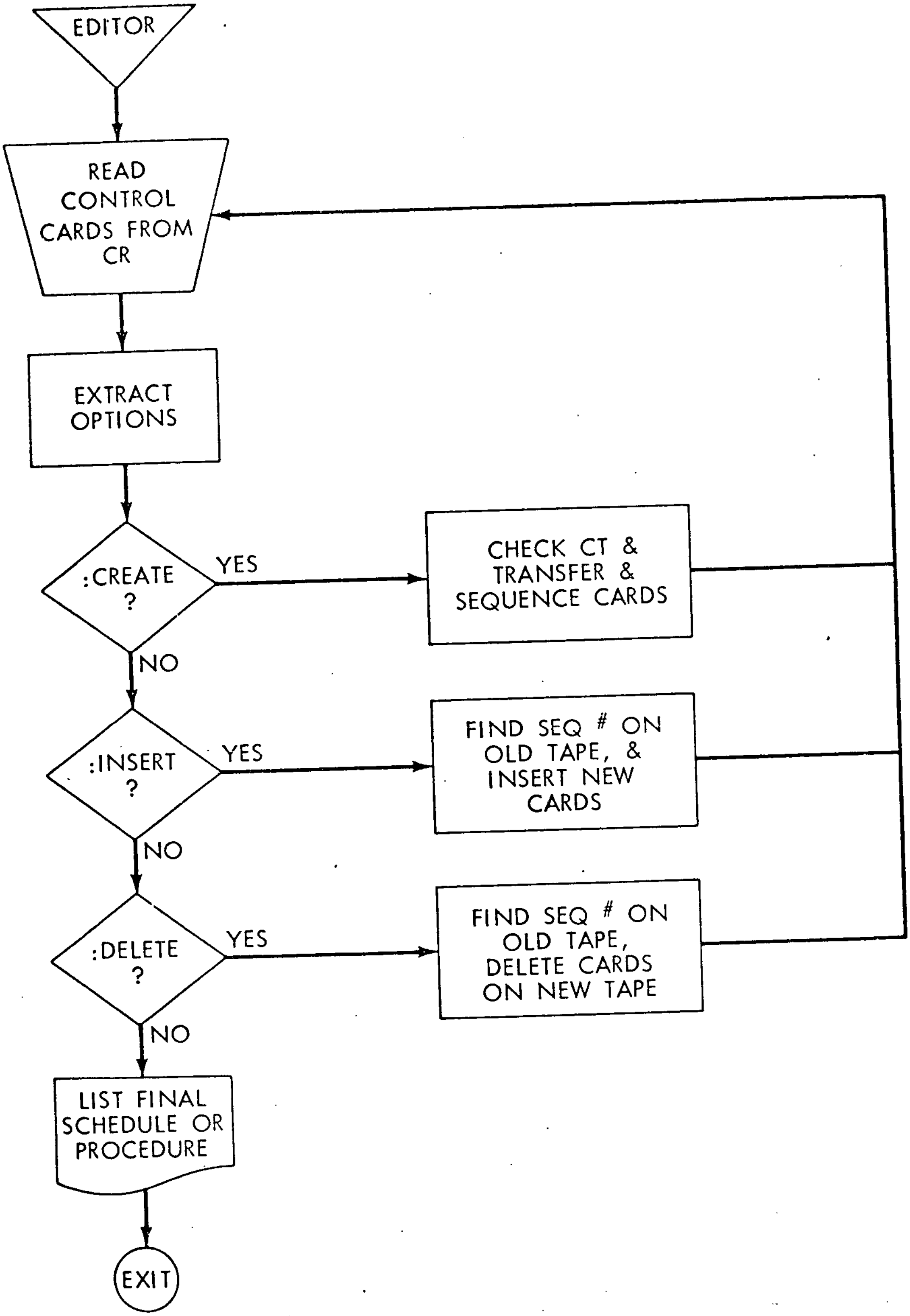

EDITOR
READ CONTROL CARDS FROM CR
EXTRACT OPTIONS
:CREATE ?
YES
NO
CHECK CT & TRANSFER & SEQUENCE CARDS
:INSERT ?
YES
NO
FIND SEQ # ON OLD TAPE, & INSERT NEW CARDS
:DELETE ?
YES
NO
FIND SEQ # ON OLD TAPE, DELETE CARDS ON NEW TAPE
LIST FINAL SCHEDULE OR PROCEDURE
EXIT

Figure 16. Basic Logic Flow of EDITOR

EDITOR sequences all statements and copies them onto the RAD for a Procedure or onto the CT (Control Tape) for a Schedule. The sequence numbers are of the form NNNNxxxx, where NNNN is the first four characters of the name used, and xxxx is a decimal sequence number.

2. EDITOR Control Cards

All EDITOR control cards are identified by a colon (:) in the first column of the card. The comment field of each EDITOR card begins in column 45.

(1) :CREATE,XXXX,NAME

XXXX - { SCHD - to create a schedule
PROC - to create a procedure }

NAME - The name of the schedule or procedure being created.

When EDITOR reads a :CREATE card, it assumes that all cards in the card reader between the :CREATE and the next :END card are the statements of the schedule or procedure being created. EDITOR sequences these statements, and writes them to the RAD if a procedure is being generated or to the CT tape if a schedule is being made.

(2) :INSERT,NAMESEQ#

NAMESEQ# - The first 8 characters of the sequence number of the schedule statement AFTER which cards are to be inserted. A special NAMESEQ# - NAMEBEFO - is used to insert statements BEFORE the first schedule statement.

When the :INSERT card is read, EDITOR copies the schedule statements on the CT tape up to and including the schedule statement NAMESEQ#. EDITOR inserts into the schedule the statements between the :INSERT card and the next EDITOR control card.

(3) :DELETE,NAMEFIRS,NAMELAST

NAMEFIRS - The first 8 characters of the sequence number of the first schedule statement to be deleted.

NAMELAST - The first 8 characters of the sequence number of the last statement to be deleted.

When EDITOR reads the :DELETE card, it copies the schedule statements on the CT tape up to but not including NAMEFIRS, and skips the schedule statements NAMEFIRS through NAMELAST.

The :INSERT and :DELETE cards can be stacked as long as the numeric portion of the sequence numbers are ascending. The entire modification stack must be terminated with a :END card which causes EDITOR to copy the remainder of the schedule.

(4) :END

The :END card is used to separate stacks and delimit procedure and schedules. Two :END cards terminate EDITOR.

3. Directions for Using EDITOR

EDITOR is a real-time program that can be run from the card reader or the OC device. To run EDITOR from the card reader the user must

(a) Place the RBM control cards and EDITOR commands into the card reader.

(b) Press the INTERRUPT button on the SIGMA 5 console.

(c) Type in C on the OC device. When PAUSE FGC prints on the OC device, key in FGC.

EDITOR will now read all the cards in the reader up to two :END cards, and produce a listing of the schedule or procedures being made.

To run EDITOR from the OC device, the user must

(a) Place only the EDITOR control cards and procedure (schedule) statements in the card reader.

(b) Press the INTERRUPT button on the SIGMA 5 console and type in RUN EDITOR on the OC device.

EDITOR uses at most two tape drives. The assignment of the tapes is as follows

CT (9TA80)	- Output tape if creating a schedule. If EDITOR is modifying an existing schedule CT will be the OLD schedule.
9TB81	- The output tape when modifying a schedule. The new schedule will be on this unit.

4. Examples

(1) Creating a procedure using RBM control cards

Below are the RBM and EDITOR control cards needed to generate a procedure called TURNON. The RBM control cards begin with a !. The executable statements of the procedure are between the :CREATE and :END cards. The entire deck is placed in the card reader and executed from the C device.

```
!JØB
!PAUSE FØC
!RUN FP,EDITØR
:CREATE,PRØC,TURNØN
           $START.                      SIMULATØRS
           *START.                      TELEMETRY
           *HØLD.                       UNTIL GET SYNC
           *HTSTART,3.                  RECØRD DATA ØN UNIT 3
           *CHARTØN,1,TM(37).           STRIP CHART FØR SUBCØM CØUNT
           *HØLD.
:END
:END
!FIN
```

(2) Creating several procedures using RBM TYC control

Several procedures may be stacked. Each procedure begins with a :CREATE card and ends with an :END card. Another :END card is added at the end of the card deck. The entire deck is placed in the card reader and executed via a "RUN EDITOR" key-in on the teletype.

```
:CREATE,PRØC,UTILEXER
           ?X,5001,5100,5050.
           ?D,5001,5108.
           ?R,5001=1.
           ?C,5001,5002,1.
           ?C,5001,5003,2.
           ?C,5001,5005,4.
           ?C,5001,5100,8.
           ?C,5100,5050,8.
           ?%
           ?D,5000,5108.
```

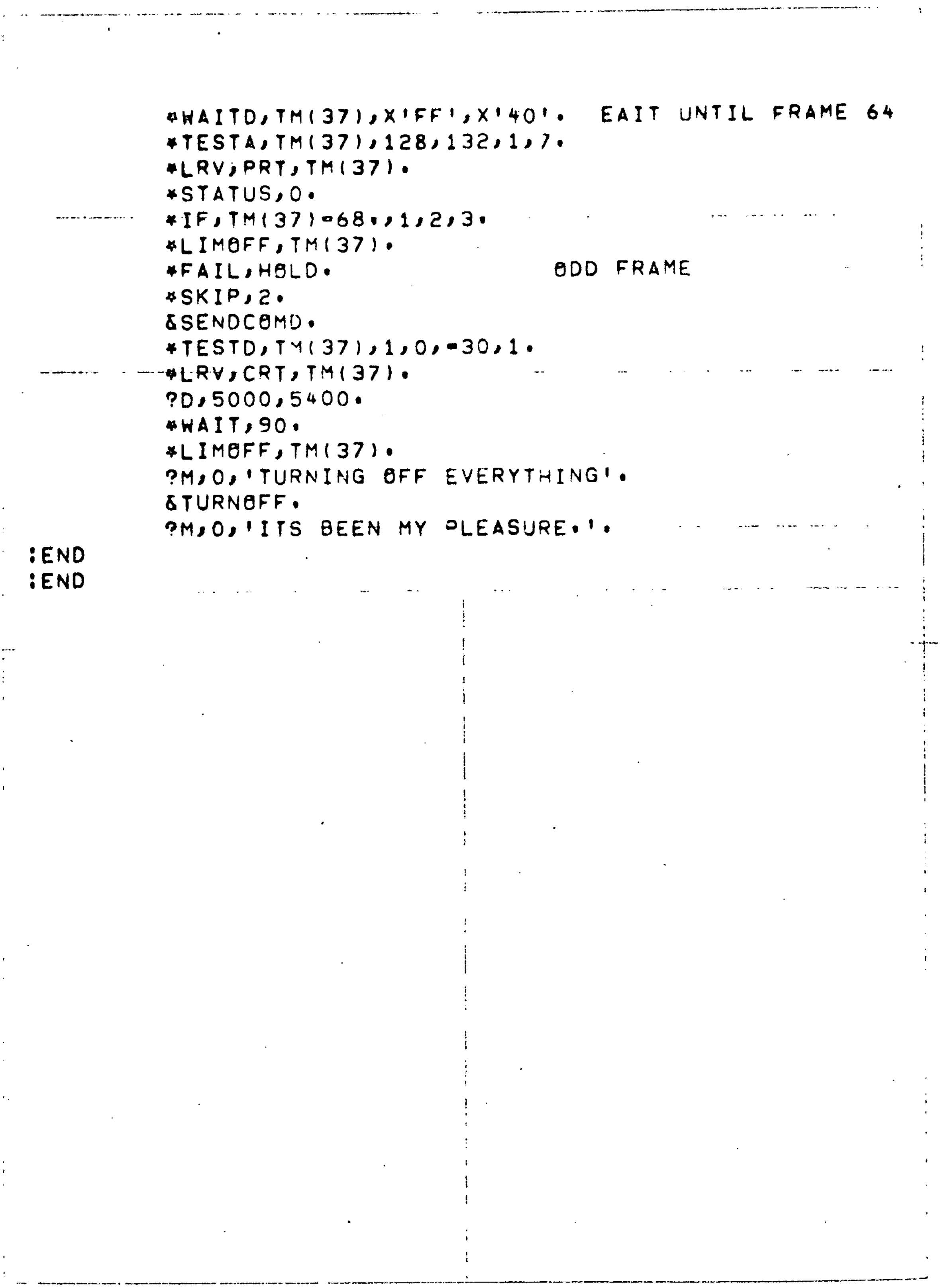
```
:END
:CREATE,PRØC,SENDCØMD
           /BIMS,VSMRC,005.
           /BIMS,ANLG3.
           /BIMS,ANLG4.
           /BIMS,ANLG5.
:END
```

```
                                                      ASSIGN CØEFICIENTS
:CREATE,PRØC,CØEFASGN
         *CØNVCØEF,TM(37),0,1.
         *CØNVCØEF,TM(9),D'1.',D'0.0',D'20.0'.              1.X**2
         *CØNVCØEF,TM(11),D'1.0',D'0.0',D'1.0'.

:END
:CREATE,PRØC,TURNØFF
         *HTSTØP.                                      STØP TELEMETRY
         *STØP.                                        SIMULATØRS
         $STØPP.                                       CHARTS
         *CHARTØFF.

:END
:END
```

(3) Creating a schedule

Below are the EDITOR control cards to generate a schedule called DEMO. DEMO calls the previously created procedures UTLEXER, COEFASGN, SENDCOMD, TURNON, and TURNOFF.

```
:CREATE,SCHD,DEMØ
         &UTILEXER.
         *HØLD.                              EVERYTHING
         &TURNØN.                            PRINT ACTIVE ARRAY &HØLD.
         $PRA.
         *HØLD.
         *LIMITS,TM(37),2,128.
         *SKIP,5.
         *LIMITS,TM(37),3,128.
         $MINØR,9,X'10',X'11',X'12'.
         *WAIT,10.
         $SWITCH.
         *WAIT,10.
         $PRA.                               8 SECØND WAIT.
         *WAIT,80.
         ?L,500A,502A,504A,506A,508A,50AA,50CA,50EA.
         *HØLD.                              SET CØEFICIENTS FØR CØNVERSIØNS.
         &CØEFASGN.                          SNAPSHØT FØR VAE
         *VAESNAP.
         *PRTCØEFS.
         *PRTLIMITS.
         *WAITA,TM(37),100,300.
         *LRV,PRT,TM(37).                    NØ VERIFICATIØ .
         *VERIFY,0.                          SEND
         &SENDCØMD.                           SET AGAIN IF ØDD FRAME.
         *TESTD,TM(37),1,1,-1,1.
         *IF,TM(37)-64.,1,2,2.               SUBCØM LT 64
         *FAIL,GØ.
         *LRV,CRT,TM(37),TM(1),TM(2).
         *PRCRT,1.
         *HØLD.
```



```
          *WAITD,TM(37),X'FF',X'40'.  EAIT UNTIL FRAME 64
          *TESTA,TM(37),128,132,1,7.
          *LRV,PRT,TM(37).
          *STATUS,0.
          *IF,TM(37)=68.,1,2,3.
          *LIMØFF,TM(37).
          *FAIL,HØLD.             ØDD FRAME
          *SKIP,2.
          &SENDCØMD.
          *TESTD,TM(37),1,0,=30,1.
          *LRV,CRT,TM(37).
          ?D,5000,5400.
          *WAIT,90.
          *LIMØFF,TM(37).
          ?M,0,'TURNING ØFF EVERYTHING'.
          &TURNØFF.
          ?M,0,'ITS BEEN MY PLEASURE.'.
:END
:END
```

VI. THE UTILITY SECTION (UTL)

UTL is a resident program consisting of several control instructions designed to facilitate the debugging of real-time programs. The instructions have a standard format, and are keyed-in through any of the CRT keyboards. These instructions are:

?L,X1,X2,X3, , , X20. — List on the requesting CRT screen the contents of the specified computer core locations. The locations are specified as hexadecimal values. Up to 20 locations can be listed.

?X,X1,X2,X3, , , X20. — Like L above, except that the CRT display is updated every 100 milliseconds with the latest contents of the specified locations.

?% — Cancel the currently operating utility function.

?D,X1,X2. — Dump the contents of core between locations X1 and X2 onto the Event Printer.

?R,A1=X1,A2=X2, , , A10=X10. — Replace the contents of locations Ai by the hex value Xi. Up to 10 replacement parameters may be used.

?C,X1,X2,N. — Copy N Sigma 5 words from the area beginning at location X1 in core to the area beginning at location X2. All parameters are in hex.

?M,N,'message' — Transmit the message to CRT #N. If N=0, the message is broadcast to all CRTs. The message appears on line 17 of the screen.

In addition to the above, a stand-alone dump routine called "DUMP" can be loaded into upper core and executed as needed. DUMP uses no system parameters and gives a complete or partial dump of core contents. If the console "I/O RESET" button is depressed before initiating DUMP, the RBM system will be re-loaded when the dump is terminated.

VII. PROGRAMMING DETAILS

This section contains information essential for designing and writing programs for the AECS system. This information will be expanded in the near future and become part of the AECS Programmers' Manual.

1. AECS Structure

AECS is a simple overlay program with two levels of overlay segments. The root (resident portion) contains all the ECS, TMD, CMD, UTL, SIM, and related programs. The first overlay level is used for directives, command programs, snapshots, and other special purpose data processing programs. The second level overlay is for CRT Page display programs. The first overlay operates at the X'6E' interrupt priority level, and the second operates at the X'6D' level. (See Figure 17.)

The steps indicated in
Figure 18 should be followed to modify an existing AECS program the steps indicated in New programs such as directives or pages can easily be added to AECS. Consider, for example, the addition of a new command program called XXXXXXXXX; the following control cards should be used:

```
!JOB    XXXXXXXXX - COMMAND PROGRAM OF NNNN EXPERIMENT
!ATTEND
!PAUSE SYC - ALLOWS RAD ACCESS
!RADEDIT
:DELETE (FILE,FP,XXXXXXXXX)
:ALLOT (FILE,FP,XXXXXXXXX), (RSIZE,30), (FORMAT,B), (FSIZE,10)
!ASSIGN (M:GO,FP,XXXXXXXXX)
!MACRSYM SI,LO,GO
          . . .
          . . .
          . . .
          END
!FIN
```

The above sequence of control cards allots space on the RAD,FP area, then compiles the program with the object deck being put on the new file XXXXXXXXX. What remains is simply to link this segment onto the rest of the AECS overlay program. The object modules are both on tape and the RAD. The job control cards for this operation are as follows:

```
!JOB AECS OVERLAY PROGRAM
!ATTEND
!PAUSE SFC
!RADEDIT
:DELETE (FILE,FP,AECS)
:ALLOT (FILE,FP,AECS), (FSIZE,1200)
!OLOAD (FILE,FP,AECS), (FORE,4A00), (PUBLIB,AELIB), (TASKS,16)
:ROOT (FILE,FP,ECS), (FILE,FP,TMD), . . . . .
:SEG (LINK 1), (FILE,FP,PAGESNAP) DIRECTIVE OVERLAY
```

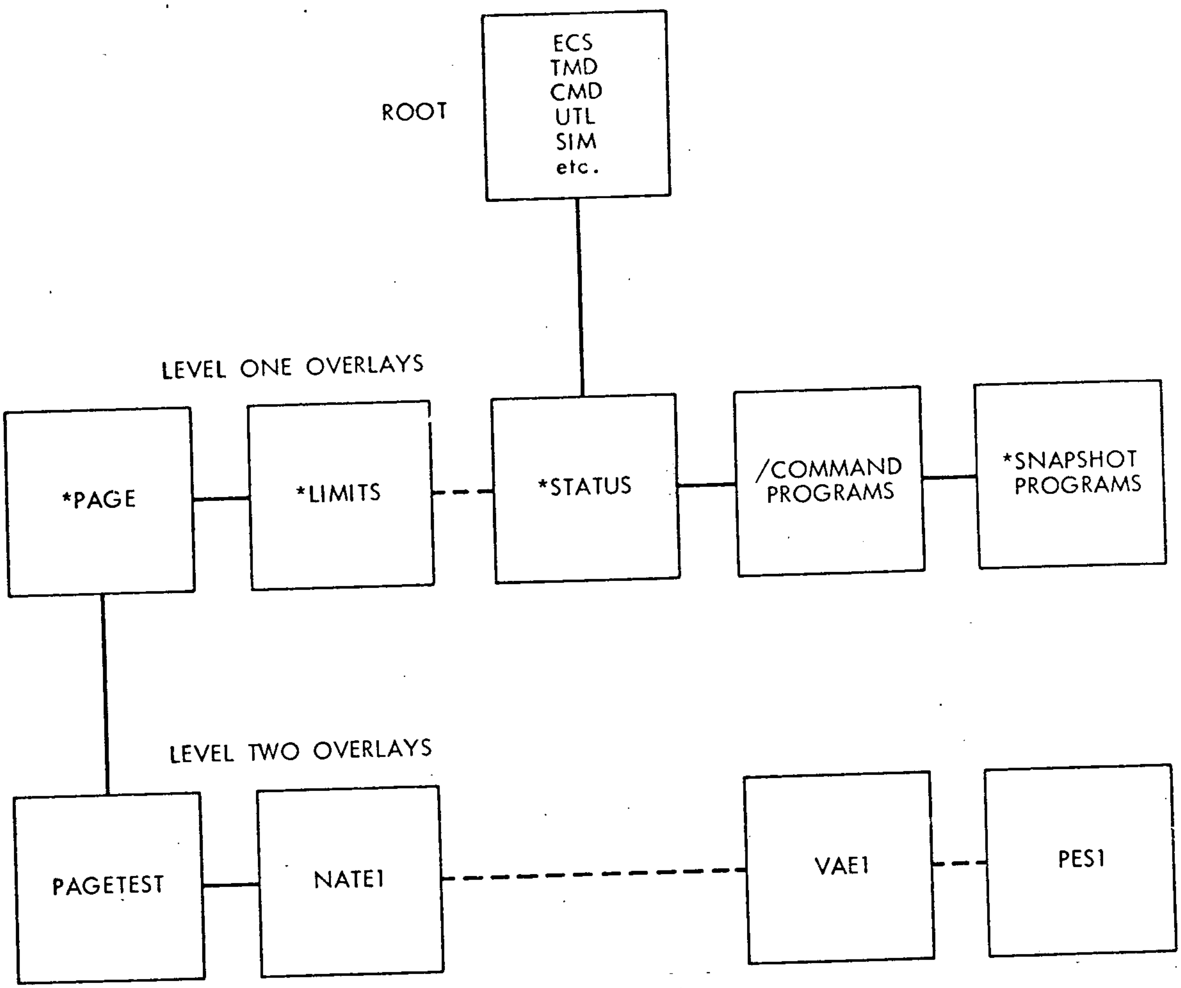


Figure 17. AECS Overlay Structure

```
:SEG (LINK,40,ONTO,1), (FILE,FP,PAGETEST)  PAGE OVERLAY
 . . . .
 . . . .
:SEG (LINK,nn), (FILE,FP,XXXXXXXXX)  NEW PROGRAM
 . . . .
 . . . .
!FIN
```

At this point, the new program XXXXXXXXX has been fully integrated into the AECS system. The system operator now can issue the call

RUN AECS

to execute the entire AECS program.

TO REPLACE A ROM TAPE PROGRAM WITH A TEST PROGRAM	REPLACE A PROGRAM ON FP AREA OF RAD	REPLACE ROM TAPE PROGRAM	RAD SAVE 9TB80	AECS ROM TAPE 9TA80	PROGRAM SOURCE
		1. SAVE SYSTEM.	OUT	—	—
		2. COPY ROM TAPE TO FP AREA OF RAD	—	IN	—
1. COMPILE PROGRAM. PUT GO FILE ON FP	1. COMPILE PROGRAM. PUT GO FILE ON FP	3. COMPILE PROGRAM. PUT GO FILE ON FP	—	—	IN
		4. COPY FP AREA OF RAD TO ROM TAPE	—	OUT	—
		5. RESTORE RAD	IN	—	—
		6. WITH SS4 ON, RELOAD SYSTEM FROM RAD. KEY IN FMEM 0	—	—	—
2. RUN AECS OVERLAY MODIFIED TO BYPASS PROGRAM ON ROM TAPE AND PICK UP TEST PROGRAM EG :SEG (LINK, XXX), (FILE, FP, TEST) :SEG (LINK, DUMY), (OPLB, BI, EOD)	2. RUN AECS OVERLAY	7. RUN AECS OVERLAY	OUT	IN	—

Figure 18. AECS Modifying Procedure

2. Directives

The following procedures are recommended for the programming of system directives:

(1) Directives are overlay segments which are called in for execution by operator key-ins, card inputs, Schedule statements, etc. As segments, the length of each directive program should not exceed 2,048 decimal words.

(2) The first word of a directive program (relative location 00000) should be the entry point — first executable instruction — of the program. ECS transfers control to this address via a

```
BAL,11    *PROGRAM
```

Control should be returned to ECS by an instruction analogous to

```
B    *11
```

(3) If there is any loop within the directive program which could last more than one second, for any reason, the variable KILLFLAG should be checked as part of the loop. That is, the following code should be added to the loop:

```
LW,2    KILLFLAG
BNEZ    EXIT - DIRECTIVE PROGRAM EXIT
```

KILLFLAG is normally 0, until set to 1 by the Cancel (%) code.

(4) If a directive program contains data which is to be updated dynamically, the first executable instruction after entry should be

```
MTW,1    UPDATE
```

where UPDATE is the flag which signals dynamic updates. If this UPDATE flag is set, the directive program is re-entered by ECS every second. The UPDATE flag is cancelled when a new directive is loaded into the system, or when the Cancel (%) code is used.

(5) All telemetry data should be obtained by LRVINDEX.

3. Page Programs

Pages are second level overlay programs. Each Page program is brought into core every 3 seconds and receives control at interrupt priority level X'6D'. The program should reference three flags, PAGEONE, PAGEUNIT, and PAGEDONE.

PAGEONE is a word (32-bit) which has an initial value of zero. It is provided so that the Page program may have some way of knowing when its first execution occurred. That is, it is up to each Page program to reset PAGEONE to some non-zero value for subsequent executions. Then a typical execution procedure could be as follows:

(1) During execution, PAGEONE is checked. If 0, this must be the first time through, so display all the static titles and header information. Also, set PAGEONE to 1 (or any other number).

(2) During execution, PAGEONE is checked. If non-zero, bypass the sending of title data and only display the dynamic telemetry data.

The advantage of this kind of operation is that I/O time to the CRT devices is thereby minimized; I/O transfer occurs at the rate of 1200 characters per second.

PAGEUNIT is another 32-bit word containing 00N00000, where N is the CRT number/screen on which this Page is to be displayed. This word needs only to be merged with the CAL2, 0 instruction for I/O to the appropriate CRT. (See Figures 18, 19, and 20.)

PAGEDONE is a resident 32-bit word which is not modified by the individual Page overlays, but which is referenced in the very last CAL2 call as the end action flag. This flag is used by the PAGE driver only.

4. Cathode Ray Tubes

The cathode ray tubes (XEROX System Keyboard Displays) on the AE system may be accessed through the statement

```
CAL2,N    CRT:FPT
B         REJECT
```

(Note: The flag CRTBUSY may be checked first; it should be 0.)

where

N - CRT unit address = 0 for all units
= 1 for CRT #1 (System CRT)
= 2 for CRT #2
= 3 for CRT #3
= 4 for CRT #4 (RCA only) etc.

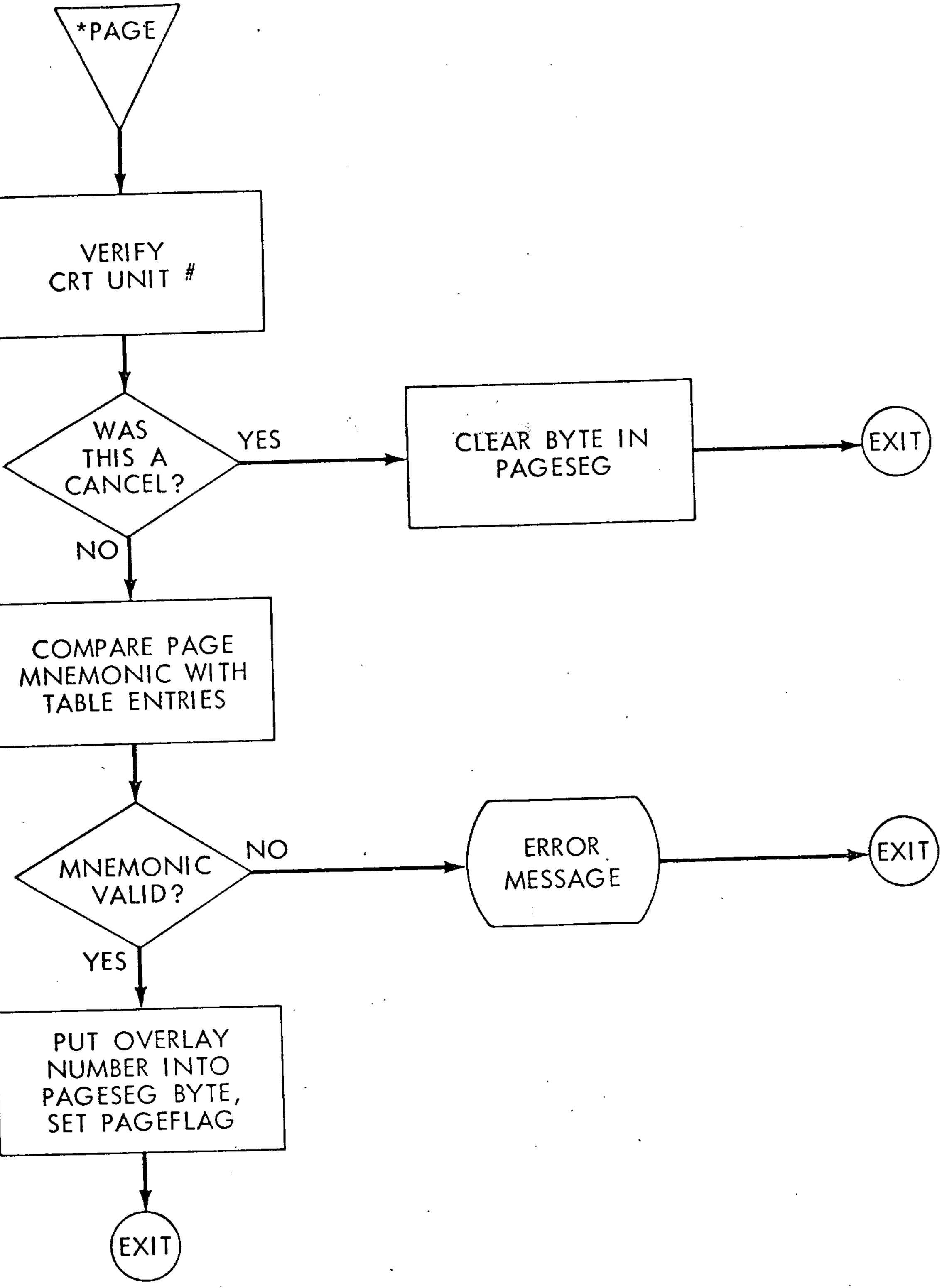


Figure 19. Flow of *PAGE Directive

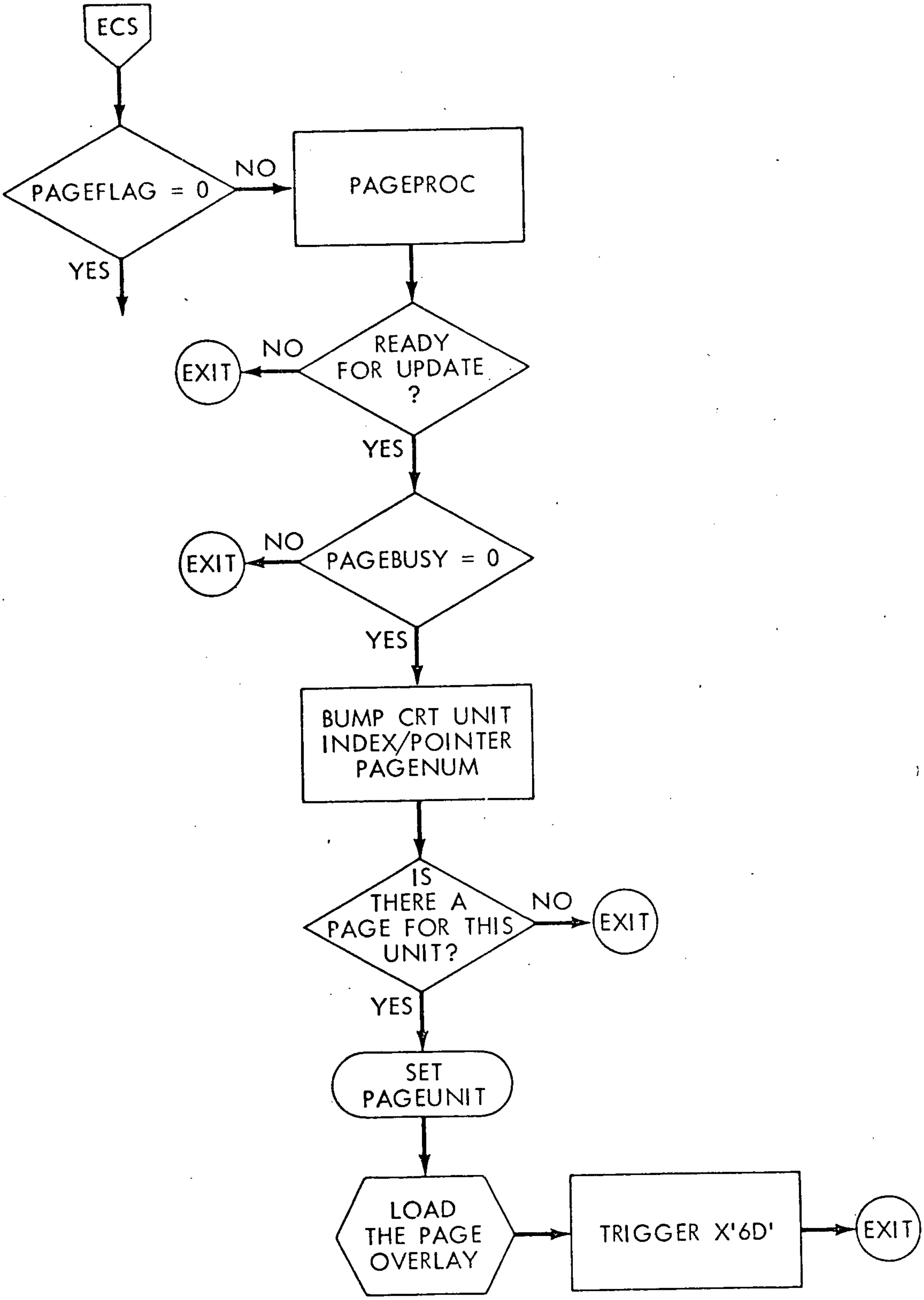


Figure 20. Flow of Resident PAGE Processor

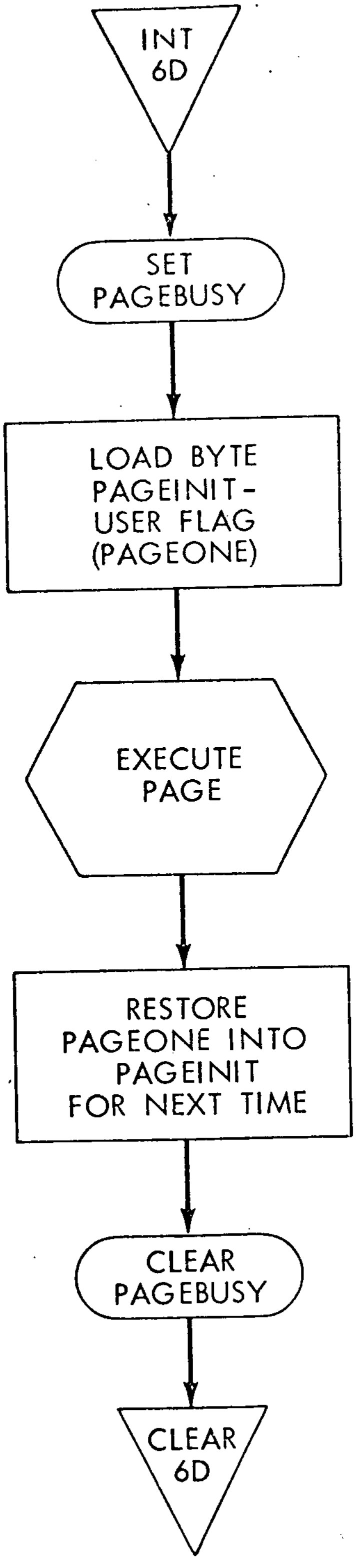


Figure 21. Flow of PAGE Interrupt/
Execution Routine

CRT:FPT - CRT function parameter table

REJECT - address of a CRT-busy routine. This address may simply be the branch, $-1

The CRT:FPT function parameter table is set up as follows:

```
            BOUND         8
CRT:FPT     GEN,8,8,16    NBYTES,CODE,TEXT
            GEN,8,8,16    LINE,COLUMN,FLAG
```

where

NBYTES — number of characters to be displayed (from 00 to 254). A full screen (17 x 80) can be displayed by specifying NBYTES = X'FF' and LINE=COLUMN=1

CODE — 0 for normal write to screen

1 for write with blinking characters

TEXT — address of text message to be displayed (TEXT is expected in EBCDIC form)

LINE — line number of CRT at which message will begin

COLUMN — column at which message will begin

FLAG — the location to store the status of the CAL2 write. -1 indicates that the CAL2 was not accepted because the I/O queue was full. Ø indicates the CAL2 was accepted but the I/O is not complete. 0 indicates that the I/O is complete. This flag must be PAGEDONE for Page overlay programs.

5. Event Printer

The Event Printer (LPA02) is accessed through the following statement:

```
CAL3,1    EP:FPT
B         $-1       Reject
```

or

```
LW,2      EPBUSY    Busy flag
BNEZ      $-1       Wait loop
CAL3,1    EP:FPT
B         $-1
```

where

EPBUSY — Event Printer busy flag

EP:FPT — Function parameter table, which is set up as follows:

```
                  BOUND         8
        EP:FPT    GEN,8,8,16    NBYTES,CODE,TEXT
                  DATA          0
```

where

NBYTES — number of characters to be printed (maximum 88)

CODE — printer format code (e.g., X'40' of X'00' for normal print, X'F1' for top of page before printing, etc.)

TEXT — address of text message to be printed

The resulting printed message appears left-adjusted on the Event Printer, followed by status information including the GMT, ~~set~~ SET, identification of statement generation device, etc.

6. Snapshot Printer

The Snapshot Printer (LPA02) may be accessed through the CAL3 statement

```
          CAL3,2     SN:FPT
          B          REJECT
```

or

```
          LW,2       SNBUSY
          BNEZ       $-1          Wait until ready
          CAL3,2     SN:FPT
          B          $-1
```

where

SN:FPT is the address of the Snapshot Printer function parameter table, and SNBUSY is the printer busy flag, which is 0 if the printer is not busy. The function parameter table is similar to that for the Event Printer, except for the GMT word:

```
                BOUND          8
SN:FPT          GEN,8,8,16     NBYTES,CODE,TEXT
                GEN,8,8,16     0,0,ENDACT
```

where

ENDACT is the address of an end-action routine, or zero. The end-action routine is entered via a BAL,11 ENDACT at the X'6C' priority level. The Snapshot Printer line contains 132 characters.

7. Spacecraft Event Printer

The Spacecraft Event Printer (LPB02) is accessed through the following statement:

```
          CAL3,3     SP:FPT
          B          $-1          Reject
```

or

```
          LW,2       SPBUSY       Busy flag
          BNEZ       $-1

          CAL3,3     SP:FPT
          B          $-1          Reject
```

where

SPBUSY — Spacecraft Event Printer busy flag

SP:FPT — Function parameter table which is identical to the Event Printer function parameter table

8. Command Calls

In order to transmit commands to the spacecraft/simulator, the following procedure is used:

(1) Put NNOOAAAA into Register 2, where NN=number of 64-bit commands to be sent, and AAAA=address of the basic command list, which must

begin on a doubleword boundary. Each entry in the basic command list is

```
                    MMOOOCCC    DDDDDDDD
```

where

MM — Minor Mode X'03' or Major Mode X'00', X'01' or X'02'

CCC — 9-bit op code

DDDDDDDD — 32-bit Minor Mode Data

(2) Execute the instruction BAL,15 COMMAND

(3) On return from COMMAND, Register 2 contains the number of commands which have been verified.

9. LRVINDEX

The subroutine LRVINDEX may be used to obtain the address of a desired telemetry word from the LRV table. Its usage is as follows:

(1) Load Register 2 with the telemetry word index (1 through 128).

(2) Load Register 3 with the sub com step index, if any.

(3) Execute the instruction BAL,15 LRVINDEX

(4) On return, Register 2 contains the byte offset desired

(5) On return, Register 3 contains an error code, if non-zero.

(6) Fetch the desired telemetry value by the instruction

```
                    LB,2    LRV,2
```

10. CONVERT

The subroutine CONVERT is used to convert an integer number ($0 \leq N \leq 2^{24} - 1$) to engineering units using existing polynomial conversion assigned to a telemetry word. CONVERT allows the user to specify the integer value or use a telemetry value from LRV. The calling sequence is:

```
(1)  LW,R2                WORD1
     LW,R3                WORD2
     BAL,R15              CONVERT
                 •
                 •
                 •
     WORD1 GEN,16,16      ADDRINT,I
     WORD2 GEN,32         J
```

where

ADDRINT — if nonzero, is the address of the integer word supplied by the user.

— if zero, then the telemetry word TM(I,J) is used as the integer value

I — telemetry word index

J — telemetry sub com step number. J=0 is used to indicate no sub com step.

(2) On normal return

R2 = floating point converted value P(X)

R3 = TM(I,J) or the user specified input.

(3) On abnormal return

R2 = The input value X or TM(I,J).

R3 = -1 to indicate no conversion polynomial for TM(I,J).

= -2 to indicate bad I,J combination

11. CONVERT2

Subroutine CONVERT2 is used to convert a floating point number to engineering units using a polynomial assigned to a specified telemetry word. The calling sequence is:

```
(1)  LW,R2                WORD1
     LW,R3                WORD2
     BAL,R15              CONVERT2
                 •
                 •
                 •
     WORD1 GEN,16,16      ADDRFP,I
     WORD2 GEN,32         J
```

where

ADDRFP — the address of the floating point number

I — telemetry word index to which the polynomial is assigned

J — telemetry sub com step number to which the polynomial is assigned. J=0 is used to indicate no sub com step number.

(2) On normal return

R2 = floating point converted value P(X)

R3 = floating point value X

(3) On abnormal returns

R2 = floating point input value X

R3 = -1 (integer) indicating no polynomial assigned to TM(I,J).

= -2 (integer) indicating a bad I,J combination

12. FORMDEC

The subroutine FORMDEC is used to convert a normalized floating point number to EBCDIC under a standard Fw.d format. The calling sequence is:

```
(1) BAL,R15         FORMDEC
    DATA            CHARRET
    DATA            =W
    DATA            =D
    DATA            FPNUM
```

where

CHARRET — is the word address of the buffer used to receive the character conversion of FPNUM.

W — is the field width in bytes. CHARRET must be large enough to receive at least W characters. $1 \leq W \leq 12$

D — is the number of characters after the decimal point. $0 \leq D \leq 8$.

FPNUM — is the address of the floating point number.

(2) On normal return the first W characters of CHARRET contain the character representation of FPNUM.

(3) On abnormal returns, the first W characters contain*.

EX: To convert the floating point number -1.5 to character under a F6.2 format and store the characters in OUTP.

```
                BAL,R15      FORMDEC
                DATA         OUTP
                DATA         =6
                DATA         =2
                DATA         FPNUM
                  •
                  •
                  •
       OUTP     RES          2
       FPNUM    DATA         =FS' -1.5'
```

On return OUTP is

		-	1	.	5	0	X	X
BYTE	0	1	2	3	4	5	6	7

VIII. SYSTEM GENERATION

System generation is the process by which a software system is created on a computer in order to enable general usage of that computer. As far as AECS is concerned, system generation involves two steps: generating the XEROX RBM standard monitor, and, generating the AECS overlay program.

1. RBM System Generation

The following basic steps are involved in a typical RBM system generation — "SYSGEN". More detailed information is available in the RBM Reference Manual.

(1) Mount the following tapes on the specified tape units:

9TA80 — Binary Input (BI) of RBM, supplied by XDS

9TB80 — FORTRAN IV/H LIBRARY

9TB81 — FORTRAN IV/H COMPILER

(2) Place the SYSGEN deck in the card reader. A listing of the AE SYSGEN deck is shown in Figure 22.

(3) Set the Sigma 5 Unit Address to '080', and LOAD and RUN the computer.

(4) In response to the "INPUT DEVICE" message on the teletype, type in "9TA80".

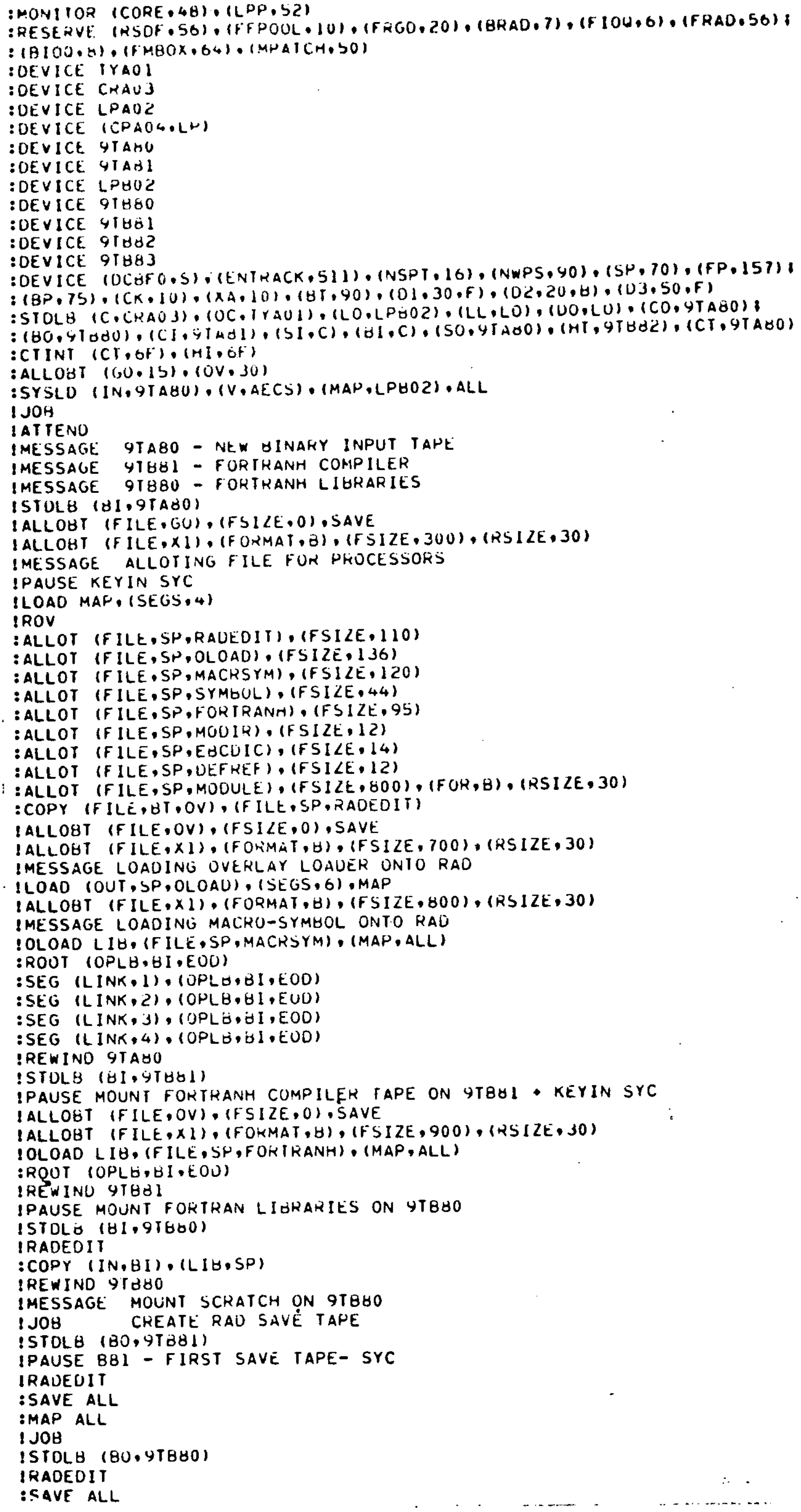
(5) In response to the teletype message "IN/OUT DEVICES?", type in the response ":SYSGEN (IN,CRA03), (OUT,LPB02)". The SYSGEN deck will then be read in from the card reader. At this point, other messages for mounting or dismounting tapes will be displayed.

2. RBM Modifications

In order to perform I/O in AECS with the CRTs, Event and Snapshot printers, the RBM X'5C' routine has been modified. The modifications to this portion of the system are listed in Figure 23.

3. AECS Overlay Program

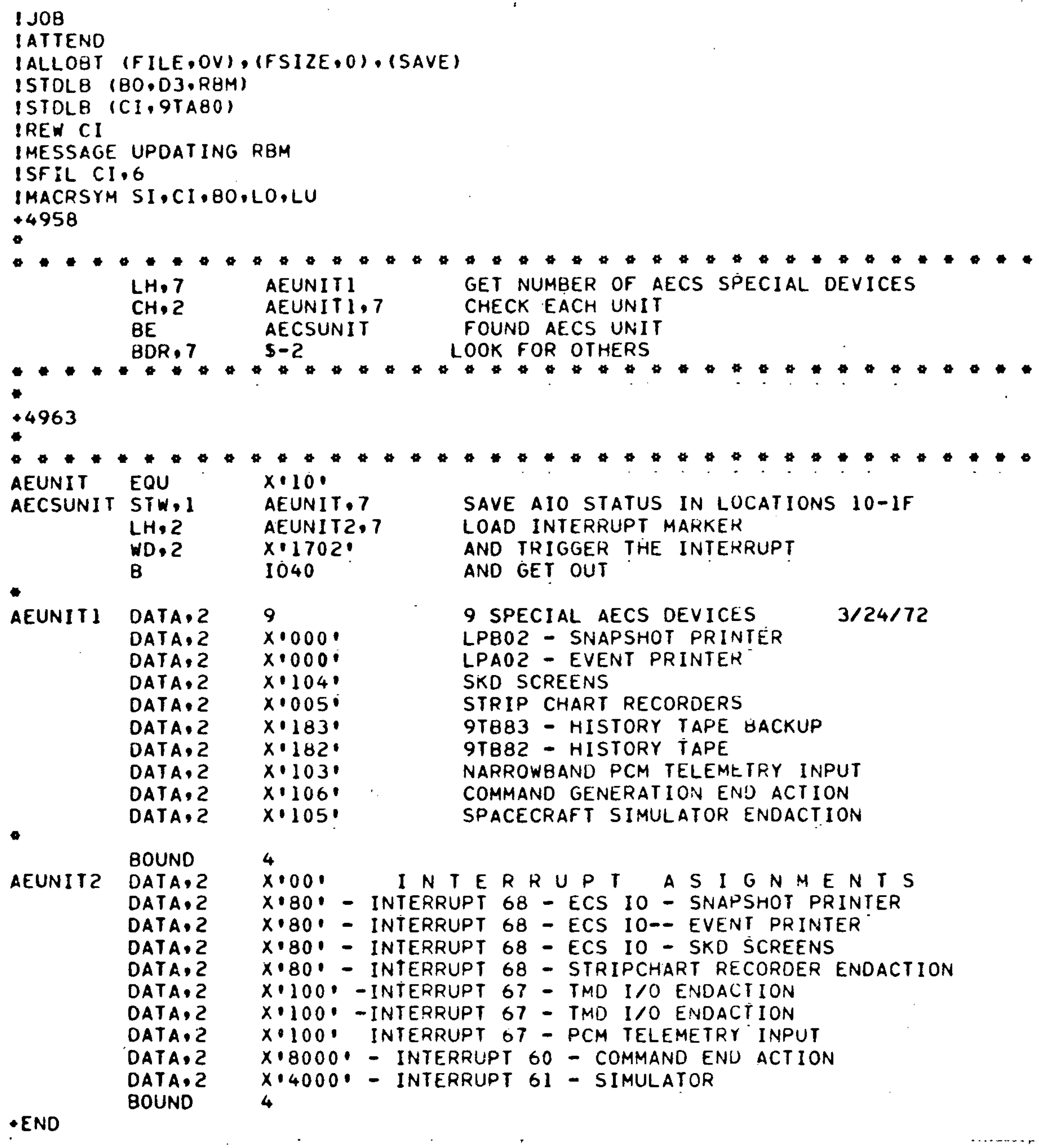
After the system is generated, the AECS overlay deck may be run. A listing of this deck is shown in Figure 24.



```
:MONITOR (CORE,48),(LPP,52)
:RESERVE (RSDF,56),(FFPOOL,10),(FRGD,20),(BRAD,7),(FIOQ,6),(FRAD,56);
:(BIOQ,8),(FMBOX,64),(MPATCH,50)
:DEVICE TYA01
:DEVICE CRA03
:DEVICE LPA02
:DEVICE (CPA04,LP)
:DEVICE 9TA80
:DEVICE 9TA81
:DEVICE LPB02
:DEVICE 9TB80
:DEVICE 9TB81
:DEVICE 9TB82
:DEVICE 9TB83
:DEVICE (DCBF0,S),(ENTRACK,511),(NSPT,16),(NWPS,90),(SP,70),(FP,157);
:(BP,75),(CK,10),(XA,10),(BT,90),(D1,30,F),(D2,20,B),(D3,50,F)
:STDLB (C,CRA03),(OC,TYA01),(LO,LPB02),(LL,LO),(DO,LO),(CO,9TA80);
:(BO,9TB80),(CI,9TA81),(SI,C),(BI,C),(SO,9TA80),(HT,9TB82),(CT,9TA80)
:CTINT (CT,6F),(HI,6F)
:ALLOBT (GO,15),(OV,30)
:SYSLD (IN,9TA80),(V,AECS),(MAP,LPB02),ALL
!JOB
!ATTEND
!MESSAGE  9TA80 - NEW BINARY INPUT TAPE
!MESSAGE  9TB81 - FORTRANH COMPILER
!MESSAGE  9TB80 - FORTRANH LIBRARIES
!STDLB (BI,9TA80)
!ALLOBT (FILE,GO),(FSIZE,0),SAVE
!ALLOBT (FILE,X1),(FORMAT,B),(FSIZE,300),(RSIZE,30)
!MESSAGE  ALLOTING FILE FOR PROCESSORS
!PAUSE KEYIN SYC
!LOAD MAP,(SEGS,4)
!ROV
:ALLOT (FILE,SP,RADEDIT),(FSIZE,110)
:ALLOT (FILE,SP,OLOAD),(FSIZE,136)
:ALLOT (FILE,SP,MACRSYM),(FSIZE,120)
:ALLOT (FILE,SP,SYMBOL),(FSIZE,44)
:ALLOT (FILE,SP,FORTRANH),(FSIZE,95)
:ALLOT (FILE,SP,MODIR),(FSIZE,12)
:ALLOT (FILE,SP,EBCDIC),(FSIZE,14)
:ALLOT (FILE,SP,DEFREF),(FSIZE,12)
:ALLOT (FILE,SP,MODULE),(FSIZE,800),(FOR,B),(RSIZE,30)
:COPY (FILE,BT,OV),(FILE,SP,RADEDIT)
!ALLOBT (FILE,OV),(FSIZE,0),SAVE
!ALLOBT (FILE,X1),(FORMAT,B),(FSIZE,700),(RSIZE,30)
!MESSAGE LOADING OVERLAY LOADER ONTO RAD
!LOAD (OUT,SP,OLOAD),(SEGS,6),MAP
!ALLOBT (FILE,X1),(FORMAT,B),(FSIZE,800),(RSIZE,30)
!MESSAGE LOADING MACRO-SYMBOL ONTO RAD
!OLOAD LIB,(FILE,SP,MACRSYM),(MAP,ALL)
:ROOT (OPLB,BI,EOD)
:SEG (LINK,1),(OPLB,BI,EOD)
:SEG (LINK,2),(OPLB,BI,EOD)
:SEG (LINK,3),(OPLB,BI,EOD)
:SEG (LINK,4),(OPLB,BI,EOD)
!REWIND 9TA80
!STDLB (BI,9TB81)
!PAUSE MOUNT FORTRANH COMPILER TAPE ON 9TB81 , KEYIN SYC
!ALLOBT (FILE,OV),(FSIZE,0),SAVE
!ALLOBT (FILE,X1),(FORMAT,B),(FSIZE,900),(RSIZE,30)
!OLOAD LIB,(FILE,SP,FORTRANH),(MAP,ALL)
:ROOT (OPLB,BI,EOD)
!REWIND 9TB81
!PAUSE MOUNT FORTRAN LIBRARIES ON 9TB80
!STDLB (BI,9TB80)
!RADEDIT
:COPY (IN,BI),(LIB,SP)
!REWIND 9TB80
!MESSAGE  MOUNT SCRATCH ON 9TB80
!JOB       CREATE RAD SAVE TAPE
!STDLB (BO,9TB81)
!PAUSE B81 - FIRST SAVE TAPE- SYC
!RADEDIT
:SAVE ALL
:MAP ALL
!JOB
!STDLB (BO,9TB80)
!RADEDIT
:SAVE ALL
```

Figure 22. Listing of AE SYSGEN Deck

Get new listing



```
!JOB
!ATTEND
!ALLOBT (FILE,OV),(FSIZE,0),(SAVE)
!STDLB (BO,D3,RBM)
!STDLB (CI,9TA80)
!REW CI
!MESSAGE UPDATING RBM
!SFIL CI,6
!MACRSYM SI,CI,BO,LO,LU
+4958
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         LH,7      AEUNIT1          GET NUMBER OF AECS SPECIAL DEVICES
         CH,2      AEUNIT1,7        CHECK EACH UNIT
         BE        AECSUNIT         FOUND AECS UNIT
         BDR,7     $-2             LOOK FOR OTHERS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
+4963
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
AEUNIT   EQU       X'10'
AECSUNIT STW,1     AEUNIT,7         SAVE AIO STATUS IN LOCATIONS 10-1F
         LH,2      AEUNIT2,7        LOAD INTERRUPT MARKER
         WD,2      X'1702'          AND TRIGGER THE INTERRUPT
         B         I040             AND GET OUT
*
AEUNIT1  DATA,2    9                9 SPECIAL AECS DEVICES       3/24/72
         DATA,2    X'000'           LPB02 - SNAPSHOT PRINTER
         DATA,2    X'000'           LPA02 - EVENT PRINTER
         DATA,2    X'104'           SKD SCREENS
         DATA,2    X'005'           STRIP CHART RECORDERS
         DATA,2    X'183'           9TB83 - HISTORY TAPE BACKUP
         DATA,2    X'182'           9TB82 - HISTORY TAPE
         DATA,2    X'103'           NARROWBAND PCM TELEMETRY INPUT
         DATA,2    X'106'           COMMAND GENERATION END ACTION
         DATA,2    X'105'           SPACECRAFT SIMULATOR ENDACTION
*
         BOUND     4
AEUNIT2  DATA,2    X'00'     I N T E R R U P T   A S I G N M E N T S
         DATA,2    X'80' - INTERRUPT 68 - ECS IO - SNAPSHOT PRINTER
         DATA,2    X'80' - INTERRUPT 68 - ECS IO-- EVENT PRINTER
         DATA,2    X'80' - INTERRUPT 68 - ECS IO - SKD SCREENS
         DATA,2    X'80' - INTERRUPT 68 - STRIPCHART RECORDER ENDACTION
         DATA,2    X'100' -INTERRUPT 67 - TMD I/O ENDACTION
         DATA,2    X'100' -INTERRUPT 67 - TMD I/O ENDACTION
         DATA,2    X'100'  INTERRUPT 67 - PCM TELEMETRY INPUT
         DATA,2    X'8000' - INTERRUPT 60 - COMMAND END ACTION
         DATA,2    X'4000' - INTERRUPT 61 - SIMULATOR
         BOUND     4
+END
```

Figure 23. Modifications to RBM X'5C'

~~Get new listing~~

```
!JOB       AECS OVERLAY
!PAU SFC
!ATT
!RADEDIT
:DELETE (FILE,FP,AECS)
:TRUNCATE FP
:SQUEEZE FP
:ALLOT (FILE,FP,AECS),(FSIZE,650),RF
!MESSAGE - RLS AECS, MOUNT 91880 FOR RAD DUMP
!OLOAD (FILE,FP,AECS),(FORE,6000),(PUBLIB,AELIB),(TASKS,16),(MAP,ALL)
:ROOT (FILE,FP,ECS),(FILE,FP,DECODE),(FILE,FP,UTILITY),(FILE,FP,TMD);
:(FILE,FP,CRT),(FILE,FP,SIM),(FILE,FP,SCHDPROC),(FILE,FP,COMMAND);
:(FILE,FP,SRENDACT),(FILE,FP,CRTPAGES),(FILE,FP,DIRECTS);
:(FILE,FP,PESCAL);
:(FILE,FP,UVOVFLOW);
:(FILE,FP,VAELIM);
:(FILE,FP,UVNDRCRT);
:(FILE,FP,PESTORE);
:(FILE,FP,NATEDAT);
:(FILE,FP,RPADATA);
:(FILE,FP,ECS:IO),(FILE,FP,TMDSTRIP),(FILE,FP,UVNADIR),(FILE,FP,TM:NATE)
:SEG (LINK,1),(FILE,FP,PAGESNAP)        PAGE,SNAP,GRIMES
:SEG (LINK,40,ONTO,1),(FILE,FP,PAGETEST)
:SEG (LINK,50,ONTO,1),(FILE,FP,UVNO1)
:SEG (LINK,51,ONTO,1),(FILE,FP,VAE1)
:SEG (LINK,56,ONTO,1),(FILE,FP,NATE1)
:SEG (LINK,64,ONTO,1),(FILE,FP,PES1)
:SEG (LINK,70,ONTO,1),(FILE,FP,RPA1)
:SEG (LINK,2),(FILE,FP,STATUS)
:SEG (LINK,3),(FILE,FP,START)
:SEG (LINK,4),(FILE,FP,STOP)
:SEG (LINK,5),(FILE,FP,HTSTART)
:SEG (LINK,6),(FILE,FP,LRV)
:SEG (LINK,7),(FILE,FP,PRCRT)
:SEG (LINK,8),(FILE,FP,SCHEDULE)
:SEG (LINK,9),(FILE,FP,HOLD)
:SEG (LINK,10),(FILE,FP,LRVSET)
:SEG (LINK,11),(FILE,FP,LIMITS)
:SEG (LINK,12),(FILE,FP,WAIT)
:SEG (LINK,13),(FILE,FP,RAW)
:SEG (LINK,14),(FILE,FP,VERIFY)
:SEG (LINK,15),(FILE,FP,DANGER)
:SEG (LINK,16),(FILE,FP,PRTCOEFS)
:SEG (LINK,17),(FILE,FP,GOTO)
:SEG (LINK,18),(FILE,FP,PRTLIMIT)
:SEG (LINK,20),(FILE,FP,TESTD)
:SEG (LINK,21),(FILE,FP,WAITD)
:SEG (LINK,22),(FILE,FP,SKIP)
:SEG (LINK,23),(FILE,FP,HTSTOP)
:SEG (LINK,24),(FILE,FP,WAITA)
:SEG (LINK,25),(FILE,FP,TESTA)
:SEG (LINK,26),(FILE,FP,DSS)
:SEG (LINK,27),(FILE,FP,FAIL)
:SEG (LINK,28),(FILE,FP,CONVCOEF)
:SEG (LINK,29),(FILE,FP,IF)
:SEG (LINK,30),(FILE,FP,DIRECT)
:SEG (LINK,31),(FILE,FP,CHARTON)
:SEG (LINK,32),(FILE,FP,CHARTOFF)
:SEG (LINK,33),(FILE,FP,LIMON)
:SEG (LINK,34),(FILE,FP,UVNO)
:SEG (LINK,35),(FILE,FP,VAE)
:SEG (LINK,36),(FILE,FP,PRUCLEAR)
:SEG (LINK,37),(FILE,FP,CONTROL)
:SEG (LINK,38),(FILE,FP,BIMS)
:SEG (LINK,39),(FILE,FP,VAESNAP)
:SEG (LINK,41),(FILE,FP,PES)
:SEG (LINK,42),(FILE,FP,RPA)
:SEG (LINK,43),(FILE,FP,MIMS)
:SEG (LINK,44),(FILE,FP,NATE)
:SEG (LINK,45),(FILE,FP,NACE)
:SEG (LINK,46),(FILE,FP,UVNOSNAP)
:SEG (LINK,47),(FILE,FP,DEFSTD)
:SEG (LINK,48),(FILE,FP,STDOSS)
:SEG (LINK,49),(FILE,FP,CEP)
:SEG (LINK,52),(FILE,FP,UVNONADR)
:SEG (LINK,53),(FILE,FP,STARTDAT)
:SEG (LINK,54),(FILE,FP,VAECT)
:SEG (LINK,55),(FILE,FP,PESSNAP)
:SEG (LINK,57),(FILE,FP,NATEON)
:SEG (LINK,58),(FILE,FP,UVNOFLOW)
:SEG (LINK,59),(FILE,FP,VAETIME)
:SEG (LINK,60),(FILE,FP,CURVTEST)
:SEG (LINK,61),(FILE,FP,LIMOUT)
:SEG (LINK,63),(FILE,FP,PESCALIB)
:SEG (LINK,65),(FILE,FP,STOPDAT)
:SEG (LINK,66),(FILE,FP,PESSTORE)
:SEG (LINK,67),(FILE,FP,EVMARKON)
:SEG (LINK,68),(FILE,FP,EVMARKOF)
:SEG (LINK,69),(FILE,FP,NATESNAP)
:SEG (LINK,71),(FILE,FP,SIMSNAP)
:ASSIGN (F:STD,D1,STD)
:ASSIGN (F:SIM,LO)
:ASSIGN (F:ACTV,D1,ACTV)
:ASSIGN (F:PROC,D3,PROCFILE)
:ASSIGN (F:DCBRAW,D1,RAWFILE)
:ASSIGN (F:PEND,D1,PEND)
:ASSIGN (F:COEF,D3,COEFFILE)
!RADEDIT
:TRUNCATE FP
:SQUEEZE FP
:SAVE ALL
!FIN
```

Figure 24. Listing of the AECS Overlay Deck

REFERENCES

1. Preliminary Document "OSO SOFTWARE DESIGN REQUIREMENT," Code 511, dated 11/11/71.

2. XDS SIGMA 5 COMPUTER REFERENCE MANUAL - 90 09 59D

3. XDS MACRO-SYMBOL Language and Operations - 90 15 78A

4. XDS REAL-TIME BATCH MONITOR (RBM) Reference - 90 15 81D

5. XDS REAL-TIME BATCH MONITOR (RBM) User's Guide - 90 16 53A

6. XDS SYSTEM KEYBOARD DISPLAY, Pub. #988021

7. Several individuals assisted in the design of the AECS as well as in the preparation of this document, especially Mr. James McGuire of the AE Project Office:

OCC

Gardiner Hall

RCA

Phillip Brandt
Mary Theus

T & E

James Bailey
Roy Morgan
David Christofalo
Russel Beard
David Provost

AE PROJECT

James McGuire
David Haykin
Wayne Hembree

8. Individuals responsible for the programming of AECS:

A. Villasenor, Code 324
W. Mocarsky, Code 324
W. Blakeslee, Code 324

APPENDIX A

SYSTEM DIRECTIVES

System directives are statements which result in a response from the system; History Tape recording may be initiated, the telemetry input stream can be stopped, etc. All directives are processed by the ECS section, and must conform to the following format:

*NNNNNNNN,ARG1,ARG2, , ,ARG20. COMMENTS

where NNNNNNNN is the directive mnemonic, ARGi are numeric or character strings, and any statement appearing after the period is treated as comments and is not processed by the system.

The basis rules for directives are:

(1) The symbol * should appear in column 10 of card inputs

(2) NNNNNNNN and ARGi may be as long as desired, but only the first 8 characters are recognized by the system.

(3) A period must complete the directive statement.

(4) ARGi may be a telemetry word, a constant, or an expression. Values of telemetry words are taken from the LRV table.

(5) Constants have the following format capabilities:

- X'NNNNNNNN' - hexadecimal (0-9, A-F); 16 digits maximum
- D'NN.NNNN' - decimal or floating point number
- NNNNNNNN - decimal integer; 16 digits maximum
- B'NNNNNNNN' - binary number (1 or 0); 64 digits maximum
- O'NNNNNNNN' - octal (0-7); 31 digits maximum

~~(6)~~ - D'N.NNE±XX' - floating point number in scientific notation.

Any number without a prefix X,D,B, or O will be treated as an integer.

(6) ~~(7)~~ When directives are submitted to the system, they will be processed as soon as the currently executing statement finishes. If the user wishes to cancel a directive, the Cancel (%) code should be used.

Special Cancel Character (%)

The special character % is used to Cancel a currently executing directive, command, or data processing (overlay) program. Utility and SIM directive operations can also be cancelled by %. If the currently executing program is contained within a Schedule or a Procedure, a HOLD mode is entered before continuing on with the next statement in the queue.

*ASSIGN,N,TM(I1),TM(I2),...,TM(ILAST). — Assigns the specified main frame telemetry words to the strip chart N. No output to these pens will begin until the *SUPGRAF. directive is issued. The time relation between the telemetry words is preserved.

EX: *ASSIGN,1,TM(3),TM(37),TM(127). — Assigns the telemetry words TM(3),TM(37),TM(127). After the *SUPGRAF directive is issued the words will be output as follows:

1. TM(3) will be output.
2. TM(37) will be output 0.033 seconds later.
3. TM(127) will be output 0.087 seconds later.
4. TM(3) will be output 0.0039 seconds later.

*CHARTOFF,P1,P2,,,Pn. — Terminates the assignment of the specified strip chart pens on the Brush recorders. If no pens are specified all strip chart assignments are cleared. The value of Pi ranges from 1 through 26.

EX: *CHARTOFF,3,7. — Cancel the operation of pens 3 and 7.

*CHARTOFF. — Cancel all pen activity.

*CHARTON,P1,TM(I1,J1),P2,TM(I2,J2),,,. — Assigns the specified pens to the corresponding telemetry words. Digital to analog conversion is linear: a telemetry value of 0 corresponds to 0.0 volts, and a value of 255 corresponds to 4.98 volts. Speed of the recording is controlled at the recoder console.

*CHECKPEN,COUNTS. — Causes the specified counts to be output to all strip chart recorders. The system will automatically enter HOLD. COUNTS is an integer number between 0 and 255.

EX: *CHECKPEN,255. — Output 255 counts (4.98 volts) to each strip chart recorder and enter hold.

*CLEAR,N. — Clears the display screen of CRT #N. If N is not specified, the screen of the calling CRT is cleared.

*CRT(N),L,C,XXXXXXXXXXXX. — Generates new page display programs or modifies existing ones. N is the Page number, L is the line number and C is the column. The resulting page overlay is permanently saved on the RAD and can be called by *PAGE,CRT(N),X. where X is the desired screen number. Up to 20 pages (N between 1 and 20) can be created in this way. There are three *CRT statements:

TYPE 1 — *CRT(N),L,C,TEXT. — Display up to 40 characters of EBCDIC text starting on line L, column C of Page N.

TYPE2 — *CRT(N),L,C,TM(I,J),IDENT. — Display the telemetry value TM(I,J) on line L, column C of page N. IDENT contains up to 8 characters of identification. Prohibited characters are (,). Blanks are removed. The leading character must be alphabetic. The telemetry value is displayed in binary unless a conversion polynomial exists.

TYPE 3 — *CRT(N),L,C,TM(I,J),MASK,TEXT0,TE~~S~~XT1,,,,. — On line L column C of page N, display the 8 character text messages corresponding to the value of TM(I,J) masked by mask. That is if MASK = B '00011000' and bits 4 and 5 of TM(I,J) are 1 and 0 respectively, display TEXT2 (2 = B '10'). The restrictions on the text statements are the same as for IDENT.

Try to put on one line

/CONTINUE. — Causes the execution of a dangerous command. This statement is normally entered during a HOLD mode.

*CONVCOEF,TM(I,J),ARG0,ARG1,...,ARGN. — Assigns the string of coefficients to the telemetry word TM(I,J) for engineering conversion. The coefficients may be entered in integer, floating point, or scientific notation. Each *CONVCOEF statement updates the coefficient table on the RAD. The coefficients are reloaded whenever AECS is initialized. Coefficients for one telemetry word can be continued on the next statement by placing a C in the field immediately before the next coefficient.

EX: *CONVCOEF,TM(9),1,2,3,4. — Assigns the polynomial

$$P(X) = 1. + 2. * X + 3. * X^2 + 4. * X^3$$

to telemetry word TM(9).

EX: *CONVCOEF,TM(46,1),D'1.2',D'7.0E-1'.

*CONVCOEF,TM(46,1),C,D1-8.2E-2'.

Assigns the polynomial

$$P(X) = 1.2 + 0.7 * X - 0.082 * X^2$$

to the telemetry word TM(46,1).

*CONVCOEF,TM(I,J),DELETE. — Removes the coefficients for TM(I,J) from the system.

*CURVTEST,I,J. — Generates a table of raw volts versus engineering units using the conversion polynomial for the specified telemetry word TM(I,J). If J is 0 the polynomial for TM(I) is used.

EX: *CURVTEST,37,0. — Tabulates the raw volts versus engineering units using the polynomial for TM(37).

*DANGER,ALLOW. — Allows execution of one dangerous command. It is assumed that the first command following this directive is the dangerous command.

*DANGER,DISALLOW. — Forces the system to enter a HOLD mode after any attempt to execute a dangerous command. The dangerous commands are not sent unless directed to do so.

*DANGER,XXXX,OPCODE,MINORBITS. — Directs the system to ADD (XXXX=ADD) or DELETE (XXXX=DELETE) from the DANGEROUS COMMAND TABLE the command having the op-code OPCODE and the 32 bit minor mode bits MINORBITS. The field MINORBITS must be present even for major mode commands.

EX: *DANGER,DELETE,0'131',0. — Deletes the command having op-code 0'131' and minor mode field of 0. (/OAPS,VAL3ON) from the DANGEROUS COMMAND TABLE.

*DECODER,N. — Selects the spacecraft command decoder N. N must be 1 or 2. AECS initially assumes DECODER #1 when the system is loaded.

EX: *DECODER,1.

/DELETE. — Cancels execution of dangerous commands. This statement is normally issued during a HOLD mode.

*DISABLE,N,NAME. — Disables the communication unit N from command transmission. That is, after execution of this statement, unit N will not be allowed to generate commands of the form /NAME. If NAME is ALL, no commands will be allowed from the specified unit. This directive can be issued from CRT #1, the Card Reader, or the Teletype.

EX: *DISABLE,2,VAE. — Prevents CRT #2 from issuing commands to the VAE experiment.

*DISABLE,3,ALL. — Prevents CRT #3 from issuing any commands.

*ENABLE,N,NAME. — Permits unit N to send commands of the form /NAME. If NAME is ALL, then all commands can be sent from unit N. This statement can be issued from the CRT #1, the Card Reader and the Teletype.

EX: *ENABLE,2,PROG. — Allows CRT #2 to issue commands of the form /PROG.

*EVMARKOF,M1,M2,M3,...,MLAST. — Terminates the assignments of the specified event markers. If no event markers are given, all assignments are terminated.

EX: *EVMARKOF,1,3,5. — Stop event markers 1,3, and 5.

EX: *EVMARKOF. — Stops all event markers.

*EVMARKON,M,TM(I,J),N,POLARITY. — Assigns the Nth bit of the telemetry word TM(I,J) to the EVENT MARKER M. The polarity is determined by POLARITY as follows:

NOINVERT - 0 volts when the bit is 0 MAX volts when the bit is a 1.

INVERT - 0 volts when the bit is a 1 MAX volts when the bit is a 0.

The bits of the telemetry word are numbered left to right with the most significant bit being denoted as bit 1. The correspondence between the EVENT MARKER numbers and the 7935 are:

M	7935 SLOT NO.	PIN NO. 7950
1	9B	47
2	9B	46
3	9B	45
4	9B	44

M	7935 SLOT NO.	PIN NO. 7950
5	9B	34
6	9B	13
7	9B	11
8	9B	05
9	01	47
10	01	46
11	01	45
12	01	44
13	01	34
14	01	13
15	01	11
16	01	05

EX: *EVMARKON,4,TM(37),3,NOINVERT. — Assigns bit number 3 of TM(37) to the event marker 4.

*FAIL,GO or *FAIL,HOLD. — Causes the comments (following the period) to be displayed on the Event Printer and on all CRT screens. No interruption in the sequence of operations occurs if the parameter GO is used; the parameter HOLD causes the system to enter a HOLD mode.

EX: *FAIL,HOLD. DATA OUT OF LIMITS. — If this statement is executed, the comments DATA OUT OF LIMITS is displayed on all the CRT screens, the Event Printer and a HOLD mode is entered. The HOLD can be cleared only by a *GO.

*GO. — Clears the system HOLD mode and continues with the next statement in a SCHEDULE. This statement has no effect if the system is not in a HOLD, and can therefore be used as a comment card.

*GOTO,N. — Issued during a HOLD mode, this statement causes the SCHEDULE to skip to statement N. If no SCHEDULE is in operation, it is a null statement. If N is greater than 9999 or less than 0, the SCHEDULE is aborted. N is the statement number in columns 77-80 of the SCHEDULE card image.

EX: *GOTO,345. — Issued during a HOLD, the statement causes the SCHEDULE to skip to statement number 345. The HOLD is still in effect, so the operator can verify that statement 345 is the desired statement. If a mistake was made, the operator can issue another GOTO.

*GROUPCMD,NNN. — Directs the system to execute the Group Commands given by group number NNN. 0 ≤ NNN ≤ 999.

*HOLD. — Sets the system in a HOLD mode. The HOLD mode simply bypasses the automatic execution of a SCHEDULE. All statements in the system can be issued during a HOLD, but the SCHEDULE cannot be continued until the directive *GO is sent.

*HTDUMP,N(,GMTSTART(,GMTSTOP)). — Prints the contents of the History Tape on unit 9TB8N starting at GMTSTART and stopping at GMTSTOP. The explanation of the parameters is given in HTPLAY.

EX: *HTDUMP,1,100:23:30:00. — Dump the History Tape on unit 9TB81 starting at GMT 100:23:30:00 until the end of the recorded data.

*HTPLAY,N(,GMTSTART(,GMTSTOP))(,H). — Directs the system to begin playing a previously recorded History Tape and have the system treat the recorded data as real time data. The arguments within the parentheses are optional.

N — Unit number of the tape drive used (0 or 1). A 0 means the tape is located on 9TB80, a 1 means the tape is located on 9TB81. This parameter must be present.

GMTSTART, GMTSTOP — GMT value at which to start and stop the HY playback, in the form DDD:HH:MM:SS (DAY:HOUR:MINUTE:SECOND). These parameters are optional. If both are omitted, the entire tape is played back. If GMTSTART only is given, playback is from GMTSTART to the end of the tape. Since the first GMT encountered on the input card is assumed to be the starting GMT, GMTSTART must be provided if GMTSTOP is desired. However, if GMTSTOP is only desired GMTSTART may be specified as a single character zero (,0,). In this case, playback is from the beginning of the tape to the GMTSTOP value.

H — Halfspeed parameter. If an H appears in the last field, playback is performed at the 8 frame per second rate. If no parameter is specified, the 16 frame/second rate is used.

EX: *HTPLAY,0. — Play the entire tape on 9TB80.

EX: *HTPLAY,1,H. — Play the entire tape on unit 9TB81 at halfspeed.

EX: *HTPLAY,1,123:03:00:00. — Play the tape on unit 9TB81 starting at GMT 123:03:00:00 to the end of tape.

EX: *HTPLAY,1,001:00:00:00,H. — Play the tape on unit 9TB81 at halfspeed starting at GMT 001:00:00:00 to the end of tape.

EX: *HTPLAY,0,001:00:00:00,001:01:00:00. — Play the tape on unit 9TB80 from GMT 001:00:00:00 up to GMT 001:01:00:00.

EX: *HTPLAY,1,0,001:01:00:00,H. — Play the tape on unit 9TB81 starting at the beginning of the tape to GMT 001:01:00:00 at halfspeed.

*HTPLAYOF. — Terminates the HISTORY TAPE PLAY BACK currently in process.

*HTSTART,N. — Starts the History Tape recording of real time PCM data. Each record on the History Tape is 560 32-bit words long. Another *HTSTART,M. can be issued to switch tape drives from N to M without loss of data. N is either 2 or 3.

EX: *HTSTART,3. — Start History Tape recording on 9TB83.

*HTSTOP. — Stops data from being placed on the History Tape.

*IF,ARITH,NEG,ZERO,POS. — Depending on the value of ARITH, causes a branch to one of three SCHEDULE or PROCEDURE statements. This statement is similiar to a FORTRAN 3-way IF instruction.

ARITH - any arithmetic expression consisting of combinations of arithmetic operations +,-,*,and/, and telemetry words or real numbers. The telemetry words, denoted by TM(I,J), are converted if a conversion polynomial exists. Grouping by parenthesis is permitted. Up to 26 numbers are allowed in the expression.

NEG - the relative position in the SCHEDULE or PROCEDURE to which control will be transferred if ARITH is negative.

ZERO - the relative position in the PROCEDURE or SCHEDULE to which control will be transferred if ARITH is zero.

POS -- the relative position in the SCHEDULE or PROCEDURE to which control will be transferred if ARITH is positive.

EX: *IF,TM(65,4)-(6.*TM(37))/8.,-1,0,2. — If this is statement P in the SCHEDULE or PROCEDURE, the control will be transferred to statements P-1,P,P+2 if the value of the expression TM(65,4)-(6.*TM(37))/8. is negative, zero or positive, respectively.

*LIMITS,TM(I,J),LOW,HIGH. — Establishes limit values for the telemetry word TM(I,J). The parameters LOW and HIGH are specified in counts. This directive defines the limits - it does not initiate limit checking; the directive *LIMON triggers limit checking.

EX: *LIMITS,TM(17),20,200. — The lower limit for TM(17) is 20 counts and the upper limit is 200 counts. If the value of TM(17) is less than 20 or greater than 200, the message DOL TM(17) = NN appears on the event printer and the CRT screens. NN is given in decimal.

*LIMSET,TM(I,J),NAME. — Associates the specified telemetry word TM(I,J) with the experiment or subsystem NAME. The limits for the word TM(I,J) must be previously defined by the *LIMITS directive. The experiment or subsystem NAME must be in the system's tables of valid subsystems.

EX: *LIMSET,TM(46,1),VAE. — Puts the telemetry word TM(46,1) in the group limit table for VAE.

*LIMOFF,XXXX,XXXX,XXXX,...,XXXX. — Stops limit checking for the specified telemetry words and group limits. If no telemetry words, experiment name, or subsystem name are specified, all limit checking is stopped.

EX: *LIMOFF,TM(46,1),UVNO. — Stops limit checking for TM(46,1) and all limits assigned to UVNO via the LIMSET directive.

*LIMON,XXXX,XXXX,,,XXXX. — Starts the limit checking for the specified telemetry words and groups of telemetry words assigned to experiments and subsystems via the LIMSET directive. It is assumed that limits for the telemetry words have been established. If no parameters after the LIMON are specified, all limits are turned on.

EX: *LIMON,VAE,TM(5). — Turns on all limits associated with the VAE experiment via the LIMSET directive and the telemetry word TM(5).

*LIMOUT,TM(I,J),&PPPPPPPP. — Forces execution of the RAD PROCEDURE PPPPPPPP if the limits for TM(I,J) are exceeded twice in succession. The PROCEDURE PPPPPPPP must be on the RAD, and the limits previously defined by the *LIMITS directive. If an out of limits condition occurs while another PROCEDURE is in operation, the executing PROCEDURE is held in ambience and replaced by PPPPPPPP. The PROCEDURE must complete execution within 3 minutes. The PROCEDURE must complete execution within 40 seconds if another emergency PROCEDURE is awaiting execution.

EX: *LIMOUT,TM(1),&SYNC. — Whenever TM(1) exceeds its limits twice in succession, the PROCEDURE SYNC is executed.

*LRV,XXX,TM(I,J),TM(I,J),...,TM(I,J). — Displays raw and converted (if a conversion exists) data for the specified telemetry words on the CRT (XXX=CRT) or Event Printer (XXX=PRT). If another directive or command is not immediately issued, the values on the CRT are updated every second.

EX: *LRV,CRT,TM(37),TM(17,1). — The specified telemetry words will be displayed on the screen of the CRT device from which the directive was issued. If this statement is part of a Schedule or Procedure, CRT #1 receives the data.

*LRVSET,TM(I,J),X1,TM(K,L),X2,... — Inserts the given value into the designated telemetry word, providing that telemetry is not currently running. The directive is primarily used for debugging purposes.

EX: *LRVSET,TM(65,9),255,TM(37),X'13'. — Inserts the value 255 into TM(65,9) and 19_{10} into TM(37).

*PAGE,CCCCCCCC,N. — Causes the PAGE "CCCCCCCC" to be displayed on CRT #N. CCCCCCCC is the name of the PAGE. If N is omitted, the screen of the CRT issuing the directive will be used. Any page may be cancelled when CCCCCCCC = CANCEL.

EX: *PAGE,NATE1,3. — Display the PAGE NATE1 on CRT 3.

EX: *PAGE,CANCEL,3. — Cancels the PAGE on CRT #3.

&PRABORT. — Aborts the current Procedure.

*PRCRT,N. — Prints the screen image of CRT #N on the Snapshot Printer. If N is omitted, the screen of the CRT issuing the directive is printed.

*PRTCOEFS. — Prints the telemetry conversion coefficients on the Snapshot Printer. While there is no limit to the number of coefficients for each telemetry word only the first eight will be printed.

*PROCLEAR,XXXX,...,XXXX. — Clears the previously defined Procedures XXXX from the RAD. Any invalid Procedure names are ignored.

EX: *PROCLEAR,VAEINIT,VAECOEFS. — Clear the Procedures VAEINIT and VAECOEFS.

*PROCLIST. — List the names of all the Procedures available on the system.

*PRTLIMIT. — Prints the contents of the limit checking tables on the Snapshot Printer.

*RAW,N1,N2,N3,,,Nm. — Prints minor frames Ni on the Snapshot Printer in hexadecimal form. The print-out is performed once every major frame, unless another directive is issued. If Ni is omitted, the entire major frame is printed.

*SCHEDULE,NAME. — Causes the system to begin execution of the Schedule tape called NAME. The Schedule is assumed to be on the CT tape drive (nominally 9TA80) and have the name NAME. The first statement to be executed regardless of the first schedule statement is the HOLD directive.

EX: *SCHEDULE,OSSX. — Begin execution of the Schedule OSSX from the tape mounted on the CT tape drive.

*SCHEDULE,NAME,LIST. — Causes the system to list the Schedule NAME on the Snapshot Printer.

EX: *SCHEDULE,OSSX,LIST. — Lists the schedule named OSSX on the Snapshot Printer.

*SEQPRT,N,F,TM(I1,J1),...,TM(Ik,Jk). — Print the specified telemetry words under the selected format (F) at the frame rate N. The formats F are:

H - Hexadecimal

R - Raw volts (20 millivolts/count)

D - Decimal

C - Converted using the polynomial for TM(I,J).

B - Binary

*SETWAIT,CTS,CLK. — Causes the system to wait until the value of the specified clock is greater than or equal to CTS.

CLK - 1 — The 62.5 msec clock obtained by concatenating TM(65,9): TM(66,9):TM(67,9)1-7:TM(37)2-8.

4 — The 4 second clock (SET) obtained by TM(65,9):TM(66,9):TM(67,9)

CTS — Clock value to leave the wait state. CTS can be specified in octal, hex, or integer.

EX: *SETWAIT,0'10',4. —Wait until the SET is 10_8. If the value of the SET was 6_8 when the directive was issued, 8 seconds will elapse before leaving the wait state.

NOTE: If the SET time has passed before the SETWAIT is issued, the statement acts as a NOP.

*STARTDAT,XXXXXXXXX. — Initiates the telemetry stripping package associated with XXXXXXXXX. XXXXXXXXX is experiment dependent.

*STOPDAT,XXXXXXXXX. — Stops the telemetry stripping package associated with XXXXXXXXX. XXXXXXXXX is experiment dependent.

*SKIP,N. — Causes the SCHEDULE or PROC to skip to the statement N away from the currently executing statement. N is any integer. If a SCHEDULE or a PROC is not executing, the statement is a NOP.

EX:
```
/VAE,EXPON.

*TESTD,TM(46,1),B'11000000',B'10000000',1,2.

*SKIP,-2.

*HOLD.
```

The *SKIP,-2. will cause the command /VAE,EXPON to be executed.

NOTE: To minimize tape motion, all skips should be forward.

*START. — Start collecting real PCM data and store in the LRV table.

*START,SIM. — Start simulating PCM data with ramp functions and store the values in the LRV table.

*STOP. – Stop collecting PCM data.

*STORE,X(I),VALUE. – Stores the specified value VALUE in the common storage location X(I).

*SUPGFOF,N1,N2,...,NLAST. – Stops all strip chart recording to the specified strip charts provided they were started via a *SUPGRAF. If no recorders are specified, all strip chart recording is stopped.

EX: *SUPGFOF,1. – Stop strip chart 1.

*SUPGRAF. – Starts all strip chart recording specified by the *ASSIGN directive.

*TESTA,TM(I,J),HIGH,LOW,YES,NO. – Checks the value of the specified telemetry word. If the value is between HIGH and LOW (inclusive) the statement YES statements relative to this one is executed. Otherwise, the statement NO statements relative to this one is executed. HIGH and LOW are specified in hundredths of volts (20 millivolts per count). The order of HIGH and LOW is interchangeable.

EX: *TEST~~S~~A,TM(68,113),200,100,2,1. – If the value of the telemetry word TM(68,113) is between 2.0 volts and 1.0 volts (correspondingly 100 and 50 counts) the statement P+2 is executed next. Otherwise P+1 will be executed next. It is assumed that this statement was number P in the Schedule or Procedure.

*TESTD,TM(I,J),MASK,VALUE,MATCH,NOMATCH. – Performs a test on the specified telemetry word as follows:

1. Form the logical product (AND) of the MASK and TM(I,J)

2. Compare the result with VALUE

3. If identical, skip to MATCH statements relative to this one

4. Otherwise, skip to NOMATCH statements relative to this one.

EX: *TESTD,TM(37),B'00000011',B'00000010',-1,1. – Assume this is statement P in a Schedule or Procedure. Whenever bit number 7 is a one and bit number 8 is a zero, the statement P-1 will be executed. Otherwise P+1 will be executed.

*TESTL, NAME, HIGH, LOW, YES, NO. — Checks the value of the specified spacecraft word NAME. If the value is between HIGH and LOW (inclusive) the statement YES statements relative to this one is executed. Otherwise, the statement NO statements relative to this one is executed. HIGH and LOW are specified in counts. The order of HIGH and LOW may be reversed. Legal mnemonics for NAME are:

Name	Telemetry Words	
WHS1ET	TM(17:18,1)	Bit 1 defines the sign 1 = -
WHS2ET	TM(17:18,2)	Bit 1 defines the sign 1 = -
SP-IP	TM(17:18,3)	
WHS-PER	TM(17:18,4)	
ARPLOTHR	TM(65,58)1-7	
ARPUPTHR	TM(65:66,58)8-14	
MASCNADR	TM(65:66,59)8-16	
BHSET	TM(65:66,60)	
BHSPER	TM(65:66,61)	
BUT1CNT	TM(65:66,62)1-10	
BUT2CNT	TM(65:66,63)1-10	
MWABIAS	TM(65,64)1-5	
DESPNOR	TM(65:66,64)6-14	
MFS-BHS	TM(81)	
MFS-SG	TM(82)	
RSLSEQ	TM(119:120,4)1-4	
RSLCMD	TM(119:120,4)5-16	
RSLMAGN	TM(119:120,4)8-11	

EX: *TESTL, ARPUPTHR, 5, 2, 1, 0. — Assume this is statement P of a Schedule or Procedure. If the value of ARPUPTHR is 5, 4, 3, or 2, the statement P+1 is executed. Otherwise the statement P+0 is executed. This sequence is actually a WAIT loop. In this case, the Schedule or Procedure waits until bits 8-14 of TM(65:66,58) are within range.

*TPREC,XX.YY. — Controls the recording of the tape recorder playback data logging on tape unit XX.

YY = ON - initiates the logging of the data

YY = OF - terminates the logging of the data

XX = 2 digit code representing the tape unit on which the log tape is located.

B0 for tape 9TB80

B1 for tape 9TB81

EX: *TPREC,B1,ON. — Start logging the tape recorder playback data on tape 9TB81.

*TPREV,XX,YY,Z. — Initiates the reformatting of the tape recorder log tape on unit XX to a History Tape formatted tape on unit YY. If the log tape has been rewound, the parameter Z = R must be present. To terminate the reformatting process before completion the operator must type in on the teletype RLS TRAP.

EX: *TPREV,B1,B0,R. — Causes the previously recorded log tape to skip on unit 9TB81 to the end of the data. The data is then reformatted and written to tape unit 9TB80.

*VERIFY,MODE,RETRIES. — Specifies the commanding mode. MODE=0 means no command verification. Mode = 1 forces command verification. RETRIES is the number of times a command will be retransmitted if it fails to verify the first time. If the command fials to verify after the number of retries, HOLD is entered.

*WAIT,N. — Pauses the specified number of tenths of seconds.

EX: *WAIT,50. — Wait 5 seconds before continuing.

*WAITA,TM(I,J),HIGH,LOW,MAXTIME. — Waits until the telemetry word TM(I,J) is between HIGH and LOW (inclusive) before continuing. If MAXTIME is specified, checking continues until MAXTIME is reached, at which point the system enters a HOLD. MAXTIME is specified in tenths of seconds. HIGH and LOW are specified in hundredths of volts (20 counts per millivolts). The order of HIGH and LOW may be interchanged.

EX: *WAITA,TM(45),050,400,10. – Wait until TM(45) is between 0.5 volts and 4.0 volts (correspondingly 25 and 200 counts). Wait a maximum of 1 second.

*WAITD,TM(I,J),MASK,VALUE,MAXTIME. – Wait until the logical product (AND) of the specified telemetry word and MASK has the value VALUE, or until MAXTIME is exceeded (if it is specified). If MAXTIME is exceeded, the system enters a HOLD.

EX: *WAITD,TM(37),X'FF',1. – Wait until the subcom counter, TM(37), is exactly 1.

*WRITPOOL,TYPE,LISTOPT. – Directs the system to either list (LISTOPT=LIST), or build (LISTOPT not specified) the Group Command file (TYPE=GROUP) or the MATRIX Command File (TYPE=MATRIX). If the LISTOPT is not specified, the Group or Matrix command cards are read from the card reader and stored on the appropriate file.

EX: *WRITPOOL,MATRIX. – Build the MATRIX COMMAND FILE as specified by the cards in the card reader.

EX: *WRITPOOL,GROUP,LIST. – List the Group commands currently on the system.

APPENDIX B

COMMAND MNEMONICS

Each subsystem is associated with a unique command program that is responsible for interpreting all possible English-language mnemonics related to that subsystem and for producing the necessary fields of the 64-bit command. This command program then transfers control to the resident COMMAND routine for transmission and verification. The mnemonics for each command program depend on the subsystem, but in general have the following form:

/NNNN,ARG1,ARG2,,,,,ARGn.

where

NNNN - 1-4 character identification of the subsystem
ARGi - alphabetic or numeric parameters, to be interpreted by the individual command programs. Each ARGi contains 1 to 8 characters.

General Command

A general purpose command programs exists in the system for use in checkout of hardware. This program is /DIRECT,NNNN. where NNN is any string of numeric characters.

EX: /DIRECT,X'0123456789ABCDEF'. — Transmit the specified data bits through the command logic. Reserved command fields, such as the spacecraft address and poly check code, are replaced by the system. Spare bit fields are inserted with zeros.

Spacecraft Memory Commands

The following commands and directives relate to the loading of the spacecraft command memory.

#CMD FORMAT,AAAAA,BBBB,T. — Defines the commands which will be loaded into the spacecraft memories. The underlined portion of the command defines any AECS command format used to generate commands. The loads are saved on the RAD for later uplinking as one load execution.

AAAAA — The S.E.T. in decimal of command execution that is to be loaded into memory. Five characters may be specified. 0 AAAAA 16384. It should be noted that since the spacecraft clock increments once every 4 seconds, there is a lapse of 4 seconds between consecutive values indicated in this field. For example, a value of 00005 would indicate a 20-second lapse from a SET of 00000. A null field implies usage of the last specified value for AAAAA.

BBBB — Indicates the memory address in octal where the generated load commands will be stored. If omitted, the default value will be the next available memory cell. 0 ≤ BBBB ≤ 37778. The CMU will be selected depending on the value of BBBB. A SET LOAD COMMAND will be generated as part of the load data each time this argument is used.

T — Indicates the last group of commands for this load block. It must be present on the last "#" command.

/MEMLD,X. — This command directs action to be performed on a previously generated load block.

X = U - Uplink the loads on the RAD load file

X = P - Print the loads on the RAD load file.

X = R - Update ground reference image with loads on the RAD file.

/MEMDP,E. — This command will uplink commands necessary to dump all of the memory specified by E and compare this to the current ground reference image.

E = 1 or E = 2 selects memory to be dumped.

*MEMREFOV,M. — All failures in the last memory-dump-compare are transferred to the memory reference image file on the RAD.

M = 1 or 2 specifies which memory image is updated.

*DUMPRINT,XX.

XX=ON - List all commands on the following dumps.

XX=OF - List only commands that failed to compare on all following dumps.

*MINDBASE,GGGGGGGG,XXXXXXXX. — Established minor mode command data base for use in load generation.

GGGGGGGG - Experiment name.

XXXXXXXXX - 32-bit minor mode data to be used as data base for the following # commands. Leading zeros may be suppressed.

Group Commands

A group command is a set of commands associated with a number between 0 and 999. The format of each command is as follows:

Cols 1-20 - Command mnemonic and parameters

Cols 22-24 - Group number (3 characters)

Cols 29-31 - Op code (used as a reference only)

Cols 33-36 - 4 digit time in tenths of seconds to wait before transmitting the next command in the group.

Cols 41--80 - Comments

Matrix Commands

A matrix command is a command associated with a letter and number, e.g., B23. The format of each command is the same as for group commands except

Cols 22-24 - Matrix number (Lxx)

Cols 33-36 - Not used

APPENDIX G

AE TEST CONDUCTOR'S CONSOLE USER GUIDE

The Test Conductor's Console provides a means of initiating pre-determined functions of the AECS without the use of the CRTs. The user can initiate the following functions:

(1) AECS-CONTROL directives, including HOLD and GO

(2) Experiment snapshots

(3) Spacecraft snapshots

(4) Procedures

(5) Group command loading and execution

(6) Matrix command loading and execution

(7) CRT pages

(8) Printed copies of CRT displays

Communication between AECS and the Test Conductor's Console is via push buttons. AECS recognizes the panel as unit #13 (X'D') after the device has been enabled.

1. AECS-CONTROL

The AECS-CONTROL buttons cause AECS to enter a HOLD mode, leave a HOLD mode, and alter the flow of schedules and procedures. They are grouped under the CONTROL section of the Test Conductor's Console.

The function of each of these buttons are:

a. SKIP - Causes AECS to skip one schedule or procedure statement. The function is reliable only during a HOLD mode. The HOLD is not cleared.

b. REPEAT - If a *HOLD (*FAIL, HOLD) was issued by a schedule or procedure, the statement immediately before the *HOLD (*FAIL, HOLD) is executed. The HOLD mode is automatically cleared.

If a *HOLD (*FAIL,HOLD) was not issued by a schedule or a procedure, but AECS is in a HOLD as a result of a % key-in or some other condition, the last schedule or procedure statement executed is repeated. The HOLD is cleared automatically.

c. PRABORT - Aborts the active procedure.

d. TAPE - Causes AECS to start executing a schedule.

e. READY/GO - Causes AECS to clear the HOLD mode.

f. WAIT - Indicator light which is lit whenever *WAIT or *SETWAIT is executing.

g. FAIL - Indicator light which is lit whenever a *FAIL is executing.

h. ENABLE - Enables the Test Conductor's Console. AECS will not accept any input from the panel unless this button has been pushed once during operations.

i. HOLD - Causes AECS to abort the current directive and enter a HOLD. The light is lit whenever AECS is in a HOLD.

2. EXPERIMENT SNAPSHOTS

The experiment snapshot buttons are grouped on the panel under EXP SNAPSHOTS. The directives issued by these buttons are listed in the Appendix.

Pressing any other buttons in this group causes AECS to enter a HOLD.

3. SPACECRAFT SNAPSHOTS

The spacecraft snapshots are grouped under S/C SNAPSHOTS. None of the buttons are currently assigned.

4. PROCEDURE

The procedure buttons, numbered 1 through 12, are grouped under PROCEDURE. The assignment of each of these buttons is:

BUTTON	PROCEDURE NAME
1	& PANBUT01
2	& PANBUT02
↓	↓
12	& PANBUT12

Whenever any of the procedure buttons are pushed, AECS searches the RAD for the appropriate procedure. If another procedure is currently active, the procedure call will be ignored by the system.

To link a given procedure to a button XY, the procedure must be named PANBUTXY when created by EDITOR.

5. GROUP COMMANDING

Group commands can be loaded and executed from the Test Conductor's Console. The buttons and thumb wheel controlling these functions are grouped under GROUP CMD.

To load and execute group commands XYZ

(a) Dial XYZ on the thumb wheel

(b) Push the LOAD button causing the selected commands to be loaded into core

(c) Push the EXECUTE button to uplink the commands.

The operator is informed if the group commands do not exist on the RAD or if the group requested to be executed does not correspond to those loaded.

6. MATRIX COMMANDS

Matrix Commands can be loaded and executed from the Test Conductor's Console. The buttons and thumb wheel controlling this function are grouped under MATRIX CMD.

To load and execute MATRIX Command Lxy:

(a) Dial Lxy on the thumb wheel

(b) Push the LOAD button

(c) Push the EXECUTE button

The operator is informed if the MATRIX command does not exist or the MATRIX command to be executed differs from the one loaded.

7. CRT PAGES

CRT Pages can be initialized from the Test Conductor's Console. These functions are incorporated in the buttons CRT1, CRT2, CRT3, and CRT4, and the associated thumb wheel.

To display the CRT page #NM on CRT #1

(a) Dial NM on thumb wheel L

(b) Press the button CRT L

Currently only pages 1 - 20 are assigned to the panel as follows:

Thumb Wheel	Page
01	*PAGE,CRT(1)
02	*PAGE,CRT(2)
↓	↓
20	*PAGE,CRT(20)

8. PRINTER IMAGES OF CRT DISPLAYS

Hard copies of the CRT displays can be obtained by using the PRINT button on the Test Conductor's Console. There is one button for each CRT.

9. TEST CONDUCTOR'S CONSOLE BUTTON - DIRECTIVE ASSIGNMENT

GROUP	BUTTON	DIRECTIVE	
CONTROL	SKIP	*SKIP,2.	
	REPEAT	*SKIP,-1.	If *HOLD was in schedule/proc.
		*SKIP,0.	Otherwise.
	PRABORT	&PRABORT.	
	TAPE	*SCHEDULE.	

GROUP	BUTTON	DIRECTIVE
CONTROL (cont.)	READY/GO	*GO.
	HOLD	%
EXP SNAPSHOT	PES	*PESSNAP.
	OSS	*OSSSNAP.
	NATE	*NATESNAP.
	UVNO	*UVNOSNAP.
	VAE	*VAESNAP.
	ESUM	*ESUMSNAP.
S/C SNAPSHOTS	NONE ASSIGNED	
PROCEDURE	1	&PANBUT01.
	2	&PANBUT02.
	↓	↓
	10	&PANBUT10.
	11	&PANBUT11.
	12	&PANBUT12.
CRT	CRT1	*PAGE,CRT(xy),1. If 1 ≤ xy ≤ 20
		*PAGE,PAGEXY,1. If XY > 20
	CRT2	*PAGE,CRT(xy),2. If 1 ≤ xy ≤ 20
		*PAGE,PAGExy,2. If xy > 20

GROUP	BUTTON	DIRECTIVE	
CRT (cont.)	CRT3	*PAGE,CRT(xy),3.	If $1 \leq xy \leq 20$
		*PAGE,PAGExy,3.	If $xy > 20$
	CRT4	*PAGE,PAGExy,4.	If $1 \leq xy \leq 20$
		*PAGE,PAGExy,4.	If $xy > 20$
PRINT	PRINT 1	*PRCRT,1.	
	PRINT 2	*PRCRT,2.	
	PRINT 3	*PRCRT,3.	
	PRINT 4	*PRCRT,4.	

APPENDIX H

USER'S GUIDE FOR CONVERSION COEFFICIENT GENERATION

The conversion coefficient generation program LLSQ fits a curve of the form

$$y = p(x) = a_1 + a_2x + \dots + a_nx^{n-1}$$

to a given set of data points using the least squares method. LLSQ computes the coefficients of the N least squares polynomials

$$y = a_1$$

$$y = a_1 + a_2x$$

$$\vdots$$

$$y = a_1 + a_2x + \dots + a_Nx^{N-1}$$

The program is located on the file BP,LLSQ as a FORTRAN source deck. Both the RBM job control cards and the input data decks are needed to run the program. LLSQ cannot be run if AECS is active.

To run LLSQ if AECS is active

1. Press the INTERRUPT button on the SIGMA 5 console.

2. Key-in RLS AECS on the teletype.

3. Press the INTERRUPT button on the SIGMA 5 console.

4. Key-in FMEM 0

5. Place the RBM control cards and LLSQ data cards in the feed hopper of the card reader.

6. Press the INTERRUPT button on the SIGMA 5 console.

7. Key-in C

At this point the deck should be read and the coefficients listed on the LO printer.

If AECS is not active, only steps 3-7 need to be performed.

A typical LLSQ deck is as follows:

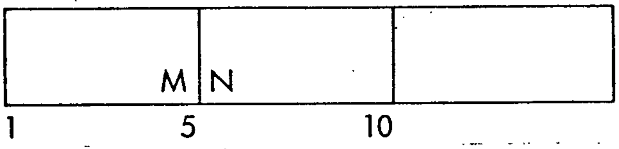
```
!JOB CONVERSION COEFFICIENTS

!ASSIGN (M:SI,BP,LLSQ)

!FORTRANH SI,GO

!OLOAD GO

!ROV

          INPUT DECK

!FIN
```

INPUT DATA DECK

The organization of the input data deck is as follows:

Parameter Card

CARD #1 - Card specifying the number of data points defining the curve to be fitted and the maximum number of coefficients to be used in the fit.

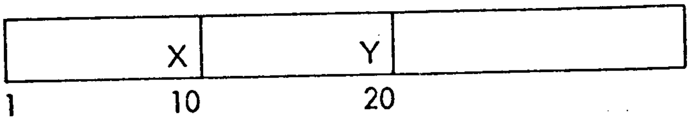


M = Integer number of data points defining the curve. The number must be punched RIGHT justified in columns 1-5.

N = Integer number specifying the maximum number of coefficients to be generated. N must be punched RIGHT justified in columns 6-10.

Coordinate Cards

The next M cards contain the data points defining the curve



X = Decimal value of the X ordinate. X must be punched in columns 1-10 and have a decimal point.

Y = Decimal value of the Y ordinate. Y must be punched in columns 11-20 and have a decimal point.

After the last coordinate card is placed a BLANK card if this is the last set of data to be processed or another CARD #1 if another set of data is to be processed.

RESTRICTIONS

1. LLSQ cannot be run if AECS is active.

2. A maximum of 15 coefficients can be generated.

3. A maximum of 100 points define the curve.